EG&G PARC

MODEL 4100

BOXCAR SYSTEM

INSTRUCTION MANUAL

220518-A-MNL-A

IMPORTANT! READ SAFETY DISCUSSIONS IN MANUAL BEFORE OPERATING INSTRUMENT

EG&G PRINCETON APPLIED RESEARCH

MODEL 4100

BOXCAR SYSTEM

INSTRUCTION MANUAL

220518-A-MNL-A

Copyright (C) 1989 EG&G INSTRUMENTS CORPORATION

Table of Contents

Chapter 1, Introduction to 4100 System 1
1.1 DESCRIPTION 1
1.2 SYSTEM CONFIGURATIONS 2
4100-1 2
4100-2 2
4100-3 2
1.3 POWER/SPACE FACTORS 2
4121B 2
4161A 2
1.4 SYSTEM CONNECTIONS 2

Chapter 2, Model 4121B 5
2.1 GENERAL 5
2.2 SPECIFICATIONS 5
2.2A INPUT CHARACTERISTICS 5
Sensitivity (full scale) 5
Maximum safe input 5
Input Coupling and Impedance 5
Gain Drift 5
DC Drift 6
Input Bandwidth 6
Input Offset 6
Dynamic Reserve 6
Overload Indicator 6
Overload Level 6
Overload Recovery 6
2.2B TIMING and CONTROL 6
Trigger Source 6
Trigger Active Indicator 6
Trigger-to-Gate Delay 6
2.2C GATE WIDTH 7
Gate Width Range 7
Gate Width Vernier 7
Gate Monitor Output 7
2.2D SAMPLING 7
Normal 7
Baseline 1 7
Baseline 2 7
Last Sample Output 7
2.2E AVERAGER 7
Techniques 7
Averager Output 7
Avg. Droop Rate 7
Averager Reset 7
2.2F MISCELLANEOUS 7
Coherent Noise vs. Delay 7
SIZE 7
Power Requirements 7
2.3 INSTALLATION 7
2.3A INCOMING INSPECTION 7
2.3B BIN REQUIREMENTS 7
2.3C GROUNDING 8
2.3D VENTILATION 8
2.4 FRONT PANEL 8
2.4A SIGNAL CHANNEL 8
2.4B GATE WIDTH 8
2.4C OUTPUT 8
Samples Averaged 8
Reset 8

AVG Connector 9
LSO Connector 9
2.4D INPUT 9
2.4E GATE MONITOR 9
2.4F TRIGGER 9
Trigger Input Connector 9
Trigger Indicator 9
Mode Selector 9
Internal 9
Delay 10
2.5 REAR PANEL 10
BNC CONNECTORS 10
RESET AVG I/O 10
BUSY OUT 10
SAMPLE VALID OUT 10
INT. OSC. OUT 10
DELAY SCAN IN 11
BASELINE I/O 11
SAMPLING MODE TOGGLE SWITCH 11
Baseline 1 11
Baseline 2 11
NIM CONNECTOR 11
2.6 USING THE 4121B 11
2.6A CONNECTIONS 11
2.6B SIGNAL SENSITIVITY 12
2.6C GATE WIDTH 12
2.6D OUTPUT 12
2.6E INPUT 12
2.6F TRIGGERING 12
2.6G BUSY OUT VS SAMPLE VALID 12
2.7 JUMPERS 12
Averager Mode 12
Delay Bypass 15
Internal Osc. Polarity 15
2.8 CIRCUIT DESCRIPTIONS 15
2.8A INTRODUCTION 15
2.8B SIGNAL INPUT 15
2.8B1 HIGH IMPEDANCE BUFFER 15
2.8B2 50 OHM DC INPUT 15
2.8C SAMPLING AMPLIFIER 15
2.8D SAMPLING BRIDGE 15
2.8E SAMPLING CAPACITORS 15
2.8F BRIDGE BALANCE CIRCUIT 15
2.8G SAMPLER RESET 16
2.8H DC OFFSET 16
2.8I Q1-Q8 DC ZERO 16
2.8J DC AMPLIFIERS 16
2.8K DC STABILIZATION 16
2.8L LSO SAMPLER 16
2.8M BASELINE SUBTRACT BUFFER 16
2.8N SIGNAL OUTPUT SAMPLER 16
2.8O AVERAGER 16
2.8P GATE WIDTH GENERATOR 17
2.8Q TRIGGER INPUT 17
2.8R FRONT PANEL GATE MONITOR 17
2.8S DELAY BOARD 17
2.8T OSCILLATOR 18
2.8U LOGIC 18
2.8U1 LSO GATING LOGIC 18
2.8U2 CALIBRATION CYCLE 18
2.8U3 START LOGIC 18

2.8U4 RESET LOGIC 18
2.8U5 NORMAL vs BASELINE LOGIC 18

Chapter 3, Model 4144 19
1.1 INTRODUCTION 19
FEATURES 19
APPLICATIONS 19
3.2 SPECIFICATIONS 20
OUTPUTS 20
T_o 20
A and B 20
T_o, A, and B PULSE LEVELS 20
A.B 20
RAMP OUT 20
PEN LIFT 20
DELAY RANGE 20
DELAY ACCURACY 20
INSERTION DELAY 20
TIMEBASE STABILITY 20
JITTER 20
TRIGGER MODES 20
External 20
Internal 20
TRIGGER RATE 20
TRIGGER INHIBIT 20
BURST MODE 20
SCAN MODE 20
SCAN INHIBIT 20
DISPLAY 21
PARAMETER SETTINGS SAVE/RECALL 21
IEEE-488 (GPIB) INTERFACE 21
DIMENSIONS 21
POWER 21
3.3 OPERATION 21
3.3A INTRODUCTION 21
3.3B FRONT-PANEL CONTROLS 21
On/Off switch 21
Display 21
Cursor 21
Keypad 21
Trigger Mode 22
Triggered Light 22
Impedance Select 22
Ext/Stop/Int Switch 22
Trigger Input Connector 22
Width Adjustments 22
T_o Connector 22
A Connector 22
B Connector 22
Pen Lift 22
Ramp Output 22
A.B 22
Trigger Inhibit 22
3.3C REAR PANEL 22
3.3D DISPLAYED FUNCTIONS 22
CH A DELAY 22
CH B DELAY 23
RATE 23
TRIGGER LEVEL 23
KEY ** TO SAVE AT ## 23
KEY ** TO RECALL ## 23

SCAN INCREMENT . . . 23
TRIGGERS/STEP . . . 23
STEPS/SCAN . . . 23
RUN SCAN . . . 23
A B AB . . . 23
T . . . 23
GPIB ADDRESS . . . 23
PEN LIFT DELAY IN/OUT . . . 23
3.3E TRIGGERING . . . 23
Introduction . . . 23
External Triggering . . . 23
Single Shot . . . 24
Triggering When Scanning . . . 24
3.3F FIXED DELAY OPERATION . . . 24
3.3G SCANNED OPERATION . . . 24
SCAN INCREMENT . . . 24
TRIGGERS/STEP . . . 24
STEPS/SCAN . . . 24
RUN SCAN . . . 25
Burst Mode . . . 25
Single vs Repetitive Scanning . . . 25
Typical Operation . . . 25
3.3H GPIB INTERFACE . . . 25
Introduction . . . 25
Code Sequences . . . 25
Saving Parameter Settings . . . 26
GPIB Serial Poll . . . 26

Chapter 4, Model 4161A . . . 27
4.1 INTRODUCTION . . . 27
4.2 SPECIFICATIONS . . . 27
Module Type . . . 27
ADC Specs . . . 27
GPIB Indicator LED's . . . 27
Digital Display . . . 27
Analog Meter . . . 27
ADC Signal Inputs . . . 27
ADC Trigger Inputs . . . 27
Reset . . . 27
Busy . . . 28
Control Interfaces . . . 28
Data Rate . . . 28
Data Format . . . 28
Bin Requirement . . . 28
4.3 INSTALLATION . . . 28
4.3A INCOMING INSPECTION . . . 28
4.3B BIN REQUIREMENTS . . . 28
4.3C GROUNDING . . . 28
4.3D VENTILATION . . . 28
4.4 FRONT PANEL . . . 28
4.4A GPIB INDICATORS . . . 28
TALK . . . 28
LISTEN . . . 28
SRQ . . . 28
REM . . . 28
4.4B DIGITAL DISPLAY . . . 28
4.4C DISPLAY SELECT SWITCH . . . 28
4.4D ANALOG METER . . . 28
4.4E ADC INPUTS . . . 28
4.4F ADC TRIGGER INPUTS . . . 28
4.4G RESET . . . 29

4.5 REAR PANEL 29
4.5A RS232C CONNECTOR 29
4.5B GPIB CONNECTOR 29
4.5C BUSY OUT 29
4.5D NIM CONNECTOR 29
4.6 OPERATION 29
4.6A INTRODUCTION 29
4.6B COMPUTER CONTROL 30
Terminator 30
GPIB DIP Switch Assembly S1 30
Commands 30
4.6C GPIB COMMAND OVERRUNS AND SERIAL POLLING 32
4.6D RS232C INTERFACE 32
RS232C PARAMETER SELECTION 32
HANDSHAKING 33
TRANSMIT/RECEIVE PIN ASSIGNMENTS 34
AVOIDING COMMAND OVERRUNS 34
LOGIC LEVELS 34
OTHER REQUIREMENTS 34
TROUBLESHOOTING RS232C 34
4.7 CIRCUIT DESCRIPTIONS 35
4.7A CLOCK GENERATOR 35
4.7B MICROPROCESSOR 35
4.7C FRONT-PANEL LED DISPLAY DRIVER 35
4.7D A/D CONVERTER 36
4.7E FRONT-PANEL ANALOG INPUT 36
4.7F TRIGGER INPUTS 36
4.7G GPIB 36
4.7H RS232 36

Appendices

Appendix A, GPIB Background 37
A.1 INTRODUCTION 37
A.2 GENERAL INTERFACE MANAGEMENT LINES 37
A.3 REMOTE MESSAGES 37

Appendix B, RS232C Background 39
B.1 INTRODUCTION 39
B.2 RS232C WORD CONSTRUCTION 39
B.3 HISTORY AND PROBLEMS OF RS232C 39

Appendix C, 4144 Interface Routine 43

Appendix D, 4161A Static Interface Routine 55
D.2 SIMPLE STATIC INTERFACE PROGRAMS 56
D.2A IBM PC/XT/AT 56
D.2B HEWLETT PACKARD 85 58

Appendix E, Pinouts 59
E.1 IEEE-488 GPIB PINOUT 59
E.2 NIM CONNECTOR 60
E.3 4161A RS232C CONNECTOR PINOUT 60

Appendix F, FCC Notice 61

Appendix G, ASCII Character Codes 63

Appendix H, Schematics 67
H.1 4121B SCHEMATICS 67
H.2 4161A SCHEMATICS 67

Index 85

Warranty . . . . . Last Page

Table of Figures

1-1, MODEL 4100 BOXCAR SYSTEM 1
1-2, 4100-3 SYSTEM CONNECTIONS 2
2-1, MODEL 4121B BOXCAR AVERAGER 5
2-2, MODEL 4121B REAR PANEL 10
2-3, MODEL 4121B SIGNAL BOARD 13
2-4, MODEL 4121B DELAY BOARD 14
3-1, MODEL 4144 DIGITAL DELAY GENERATOR 19
3-2, TIMING DIAGRAM 20
4-1, MODEL 4161A 27
4-2, BACK PANEL 29
4-3, GPIB SWITCH ASSEMBLY 31
4-4, RS232C PARAMETERS (SWITCH ASSEMBLY) 33
B-1, TYPICAL RS232C DATA WORD 40
D-1, STATIC INTERFACE ROUTINE FLOWCHART 55

Table of Tables

3-1, MODEL 4144 COMMAND SET 26
4-1, SERIAL POLL STATUS BYTE BIT WEIGHTING 32
E-1, GPIB CONNECTOR PINOUT 59
E-2, NIM PINOUT 60
E-3, RS232 CONNECTOR PINOUT 60

Chapter 1
Introduction to 4100 System

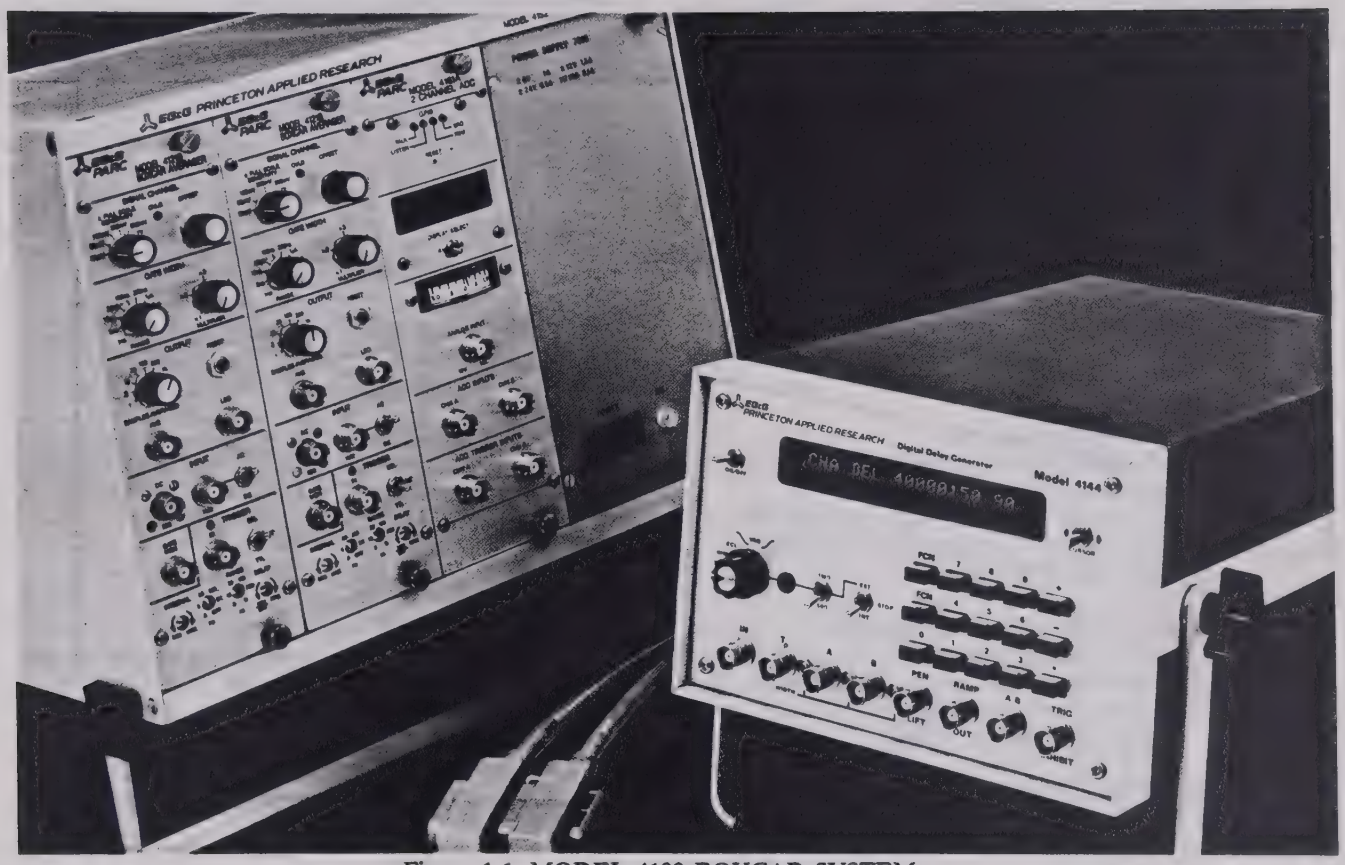


Figure 1-1. MODEL 4100 BOXCAR SYSTEM

1.1 DESCRIPTION

The EG&G Model 4100 is a series of low cost, high-performance, modularized Boxcar Integrator systems featuring ease of operation, experiment control, and outstanding performance. These systems consist of one or two samplers, a half-width NIM bin, a display module with both IEEE-488 GPIB and RS232C computer interfaces, and a separate digital delay generator, also with IEEE-488 GPIB control. A 4100 System can be controlled from its front panel, from your software, or from our exclusive Applications software. Menu driven for ease of use, this software additionally features a command mode for the advanced user's convenience. It runs on IBM PC's (including PS/2's) and true compatibles with CGA, EGA, or graphics monochrome monitor cards.

Features of the 4100 include:

- 1.5 ns gate width
- 50 Ω dc and 1 MΩ ac or dc inputs
- 400 MHz Input Bandwidth
- 20 ns Trigger to Sample Delay
- 100 kHz Trigger Rate
- Analog or Digital Delay
- Last Sample Output & Samples Averaged Output
- Baseline sampling
- Linear or Exponential Averaging
- 20 mV Sensitivity

Features of the 4144 two-channel Digital Delay Generator include:

- Delay Range of 10 ps to 100 ms
- Maximum Insertion Delay of 40 ns
- Both Fixed Delay and Scan Modes
- Jitter < 50 ps + .001% of Delay Setting

- Two independent channels
- Ramp Out for X/Y Recorder

1.2 SYSTEM CONFIGURATIONS

Three different system configurations are available, as follows.

4100-1 Boxcar System includes:
One Model 4152 Bin with Power Supply
One Model 4121B Boxcar Averager

4100-2 Boxcar System includes:
One Model 4152 Bin with Power Supply
Two Model 4121B Boxcar Averagers
One Model 4161A Display - A/D Module
Connecting BNC cables

4100-3 Boxcar System includes:
One Model 4152 Bin with Power Supply
Two Model 4121B Boxcar Averagers
One Model 4161A Display - A/D Module
One Model 4144 Digital Delay Generator
Connecting BNC cables.

The various system components are also available individually. Note that the 4152 bin includes the power supply (Model 7018). The bin can also be purchased without the power supply as the Model 4152/99.

1.3 POWER/SPACE FACTORS

Both the 4121B and 4161A modules take their power from the 4152 Power Bin. The bin can supply ±24 V @ 500 mA, ±12 V @ 1500 mA & ±6 V @ 1000 mA. The individual module requirements are:

4121B:
+24 V @ 200 mA, +12 V @ 300, +6 V @ 160
-24 V @ 150 mA, -12 V @ 590, -6 V @ 360

4161A:
+24 V @ 50 mA, +12 V @ 600, +6 V @ 550
-24 V @ 50 mA,, -12 V @ 30, -6 V @ 10

As can be quickly determined, the 4152 Bin can NOT meet the power requirements of three 4121B's or two 4161A's, even though they will physically fit. Two 4121B's and one 4161A are the most that can be operated in the 4152 bin at a time.

1.4 SYSTEM CONNECTIONS

Since the 4100-1 and 4100-2 are subsets of the 4100-3, the following discussion of the connections required by the 4100-3 should suffice for all three systems.

As depicted in Figure 1-2, assume there are two signals to be recovered. One would be applied to the Signal input of the first Model 4121B (designated the "**A**" unit) and the other to the Signal input of the second Model 4121B (designated the "**B**" unit). The trigger generated by the experiment would be applied to the Trigger **In** connector of the 4144 Delay Generator. Next, the 4144's **A** Output would be applied to the **Trigger Input** of the **A** 4121B and the 4144's **B** Output would be applied to the **Trigger Input** of the **B** 4121B.

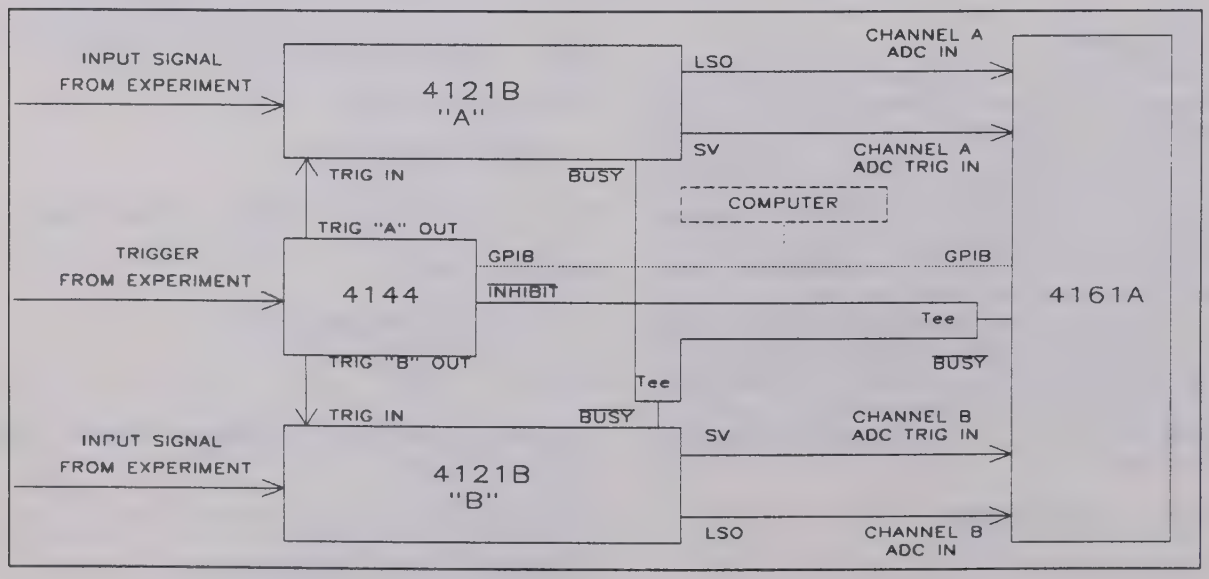


Figure 1-2. 4100-3 SYSTEM CONNECTIONS

Proper system operation requires that triggering of the 4144 be held off whenever a 4121B or the 4161A are busy. This is done by tying the **BUSY** output of the 4161A and that of each of the 4121B's to the **Trigger Inhibit** connector of the 4144. To do so, you will need two BNC **Tee**'s. In Figure 1-2, there is a Tee connection at the 4161A and at the **"B"** 4121B. It doesn't really matter where they go as long as the **Busy**'s are tied together and to the **Trigger Inhibit** input of the 4144.

The 4121B's have both **Last Sample** and **Averaged** outputs. Because the system includes a 4161A, the assumption is that each sample will be digitally converted by the 4161A and then sent to the host computer for averaging or other processing with the whole systemoperating under computer control. As a result, the **LSO** output of 4121B **"A"** is applied to the **Chn A ADC Input** of the 4161A and the **LSO** output of the 4121B **"B"** is applied to the **Chn B ADC Input** of the 4161A.

The **SV** output of each 4121B is applied to the corresponding **Trigger Input** of the 4161A. Because **SV** isn't asserted until **LSO** stabilizes, this arrangement assures that the signals convertered by the 4161A are valid.

Both the 4144 and the 4161A are controlled by the computer via the IEEE-488 GPIB interface. As shown in Figure 1-2, the host computer would normally control the experiment as well. In an application using the EG&G Boxcar Applications Software, the computer would have to be equipped with a General Instruments PC2 or PC2A GPIB Controller Card.

Chapter 2
Model 4121B

2.1 GENERAL

The Model 4121B Boxcar Averager (Figure 2-1) is a low-cost NIM plug-in module that repetitively samples an applied input signal when triggered. These samples are averaged to obtain an improvement in SNIR (Signal-to-Noise Improvement Ratio). The 4121B provides both exponential (default) and linear averaging. In exponential averaging, providing enough triggers have occurred, the number of samples averaged is that selected at the front panel (SNIR is $N^{\frac{1}{2}}$). The output (±10 V f.s.) defines the amplitude of the input signal at the sampled point. If the delay of the applied trigger is externally scanned with respect to the input signal, the sampling gate also scans, allowing recovery of the entire signal waveform.

In linear averaging, the samples simply add, and the output simply tracks the sum. The SNIR is $N^{\frac{1}{2}}$, where N is the number of triggers.

The 4121B offers a choice of gate widths from 1 ns to 150 μs with an adjustable vernier.

The applied trigger must be synchronous with the signal input to sample the same point on successive repetitions of the signal. A front-panel switch allows both external and internal triggering to be selected. If triggered internally, the oscillator output, available at a rear-panel connector, can be used to synchronize the external signal source with the Model 4121B. Front-panel controls allow the position of the sample with respect to the trigger to be set over a range of 3 ns to 1 μs.

Full-scale Input Sensitivity ranges from 20 mV to 2 V are provided to accommodate a wide range of input signals. You can select either a 50 Ω or 1 MΩ input impedance and, at 1 MΩ, either ac or dc coupling.

2.2 SPECIFICATIONS

2.2A INPUT CHARACTERISTICS

Sensitivity (full scale): ±20 mV to ±2 V in 2-5-10 sequence.

Maximum safe input:
±5 V with 50 Ω input impedance
±100 V with 1 MΩ input impedance.

Input Coupling and Impedance:
50 Ω DC, shunted by 10 pF
1 MΩ, AC/DC, shunted by 30 pF

Gain Drift:
0.5% per °C; gate width >30 ns
1.0% per °C; gate width <10 ns

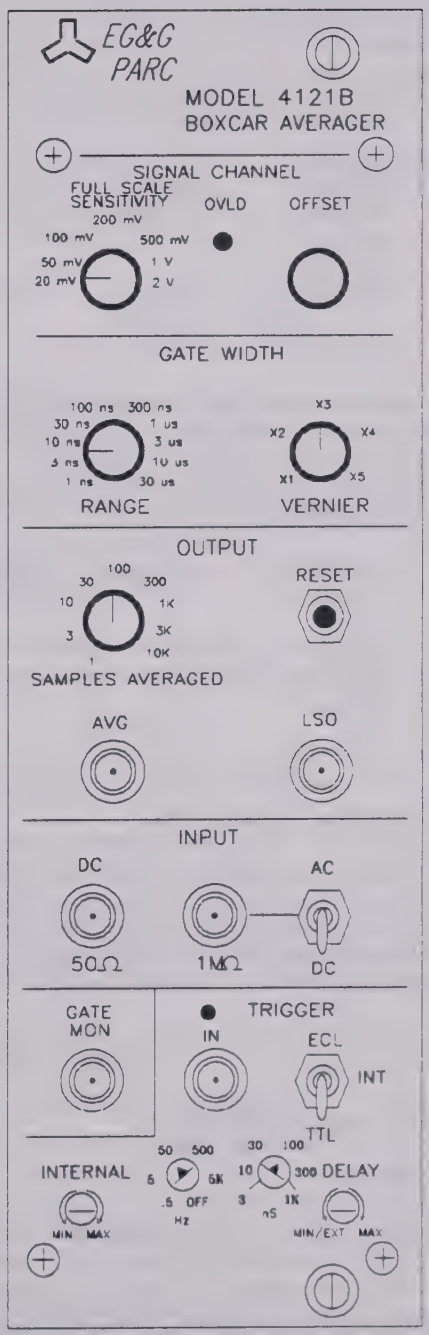


Figure 2-1. MODEL 4121B BOXCAR AVERAGER

DC Drift (referred to input):
0.2% per °C; gate width >20 ns
1.0% per °C; gate width <20 ns

Input Bandwidth:
50 Ω DC input: dc to 450 MHz.
1 MΩ AC input; 1.5 Hz to 100 MHz.
1 MΩ DC input; dc to 100 MHz.

Input Offset: Offset magnitude set by ten-turn control. Range is a function of sensitivity as follows.

SENSITIVITY	OFFSET RANGE
20 mV	200 mV
50 mV	500 mV
100 mV	1 V
200 mV	2 V
500 mV	2 V
1 V	2 V
2 V	2 V

Dynamic Reserve: This is the amount of signal that can be applied without overloading the front end. It is a function of sensitivity as follows. Indicated values are multiplier factors. To convert to volts, multiply by the corresponding sensitivity.

SENSITIVITY	DYNAMIC RESERVE
20 mV	100
50 mV	40
100 mV	20
200 mV	10
500 mV	4
1 V	2
2 V	1

Overload Indicator: Lighted LED

Overload Level: Indicator lights for both input and output overload. Input overload occurs when the input signal plus noise exceeds ±2.2 V. Output overload occurs when the output exceeds ±10 V. Transient overloads will light the indicator for a minimum of 0.1 s.

Overload Recovery: Recovers after one sample for 10 times input overload.

2.2B TIMING and CONTROL

Trigger Source:

Internal: 0.5 Hz to 5 kHz selectable with range switch. X1 to X10 multiplier gives 0.5 Hz to 40 kHz range. Oscillator output (TTL) provided at rear-panel connector. Sense of output selectable by internal jumper. The Trigger to Gate Delay (interval between negative going edge of Internal Oscillator output and positive going edge of Gate Monitor output) varies with Delay Vernier setting. Typical values follow.

SETTING	MIN	MAX
3 ns	3.5	10
10 ns	3.5	20
30 ns	3.5	38
100 ns	3.5	104
300 ns	3.5	290
1000 ns	3.5	920

External:

ECL: Positive-going edge of 5 ns minimum pulse width with termination of 50 Ω to -2 V. -5 V to +10 V safe input. 100 kHz maximum (20 kHz if baseline sampling).

TTL: Negative-going edge of 20 ns minimum pulse width. -5 V to +10 V safe input. 100 kHz maximum (20 kHz if baseline sampling).

Trigger Active Indicator: Green TRIGGERED indicator lights for at least 0.1 s each time instrument triggers.

Trigger-to-Gate Delay: Front-panel controls (switch and vernier) allow delay to be set. Switch selections are 3 ns, 10 ns, 30 ns, 100 ns, 300 ns, and 1000 ns. The selected delay is obtained with the vernier set to MAX. On the short ranges, the actual delay will be larger than that set because of the minimum intrinsic delay. Typically, the actual delay (External Triggering) on each range is as follows. Values are in ns unless otherwise stated. **Note:** As a result of a late design enhancement, the actual maximum delay on the longest range is 3 ms and not 1000 ns.

	TTL		ECL	
SETTING	MIN	MAX	MIN	MAX
3 ns	24	30	20	26
10 ns	24	40	20	36
30 ns	24	58	20	54
100 ns	24	124	20	120
300 ns	24	310	20	306
1000 ns	24	3 ms	20	3 ms

Notes:

TTL Delay: Delay was measured between the negative-going edge of the applied trigger and the positive-going edge at the Gate Monitor output.

ECL Delay: Delay was measured between the positive-going edge of the applied trigger and the positive-going edge of the Gate Monitor output.

An internal jumper can be used to bypass the delay circuits altogether to achieve the minimum possible delay. If this is done, the minimum delay with ECL triggering is typically 14 ns. With TTL triggering it is typically 16 ns.

2.2C GATE WIDTH

Gate Width Range: 1 ns to 30 ls in 1-3-10 sequence. At 1 ns the gain is reduced by not more than 50%.

Gate Width Vernier: 1 to 5 times the gate width range, one-turn control.

Gate Monitor Output: Marker pulse equal in width to gate width provided at front-panel GATE MON connector (BNC). Amplitude is 0.3 V into 50 Ω. Delay with respect to true gate position will not exceed 5 ns with 50 Ω input impedance (more with 1 MΩ input impedance).

2.2D SAMPLING

Normal: 4121B takes one sample for each trigger. All samples taken are averaged.

Baseline 1: 4121B takes one sample for each trigger. Rear-panel TTL BASELINE I/O connector provides **output** that switches synchronously with the falling edge of each TTL trigger (positive-going edge of ECL trigger). Sample taken when BASELINE I/O is high is a signal sample. Sample taken when BASELINE I/O is low is a baseline sample. The difference (each signal sample minus the following baseline sample) is averaged.

Baseline 2: 4121B takes one sample for each trigger. Rear panel BASELINE I/O connector acts as **input** for externally derived TTL control signal. Sample taken when BASELINE I/O is high is a signal sample. Sample taken when BASELINE I/O is low is a baseline sample. The difference (each signal sample minus following baseline sample) is averaged.

Last Sample Output: Analog voltage of most recent sample is provided at front-panel LSO OUT connector (BNC). Output is +10 V at 50 Ω for f.s. input, capable of driving 2 kΩ load. LSO droop rate is typically < 0.2% of output f.s. per second (< 20 mV/s).

2.2E AVERAGER

Techniques: Both exponential and linear averaging are provided. Selection is made by internal jumper. Default selection is EXPONENTIAL.

Samples Averaged: Front-panel switch selects values from 1 to 10,000 in 1-3-10 sequence.

Exponential Averaging: Normal or typical operation. Output signal at the AVG connector asymptotically approaches the steady state average as the number of samples increases. Output is at 63% when number of triggers equals SAMPLES AVERAGED setting. Output is within 1% of final value when number of triggers equals five times SAMPLES AVERAGED setting.

Linear Averaging: All samples contribute equally to the output. Output at the AVG connector increases in equal amounts with each sample of the same input signal segment until output overload occurs or the averager is reset. With a full-scale input applied, output will reach full scale when the number of triggers is 67% of the number selected by the SAMPLES AVERAGED front-panel switch.

Averager Output: ±10 V full-scale. 50 Ω output impedance capable of driving a 2 kΩ load.

Avg. Droop Rate: When not triggering, the output droop rate is a function of the SAMPLES AVERAGED switch setting as follows.

SETTING	OUTPUT DROOP RATE
1 to 30	< 1% per minute
300 to 10k	< 0.02% per minute

When triggering, error due to droop will be less than 1% of full scale for trigger rates greater than 1 Hz.

Averager Reset: Averager reset via front-panel pushbutton or by negative-going edge of TTL pulse applied to the rear-panel RESET AVG BNC connector.

2.2F MISCELLANEOUS

Coherent Noise vs. Delay: Less than ±8 mV during first 100 ns and less than ±1 mV after 100 ns. Referred to input.

SIZE: 2.70 inches W x 8.75 inches H x 10.5 inches long (6.86 cm W x 22.23 cm H x 26.67 cm L)

Power Requirements: The Model 4121B receives its power from a 4152 Power Bin.

Note: The bin without a power supply is the Model 4152/99. With the power supply installed it is simply a Model 4152. The bin's power supply is the Model 7018.

2.3 INSTALLATION

2.3A INCOMING INSPECTION

Newly received apparatus should be inspected for shipping damage. If any is noted, immediately notify EG&G PARC and file a claim with the carrier. The shipping container should be saved for inspection by the carrier.

2.3B BIN REQUIREMENTS

The Model 4121B Boxcar Averager is a double-width Nuclear Instrument Module (NIM). As such, it requires a NIM bin that conforms to the NIM standard as defined in AEC report *TID-20893 (Rev), Standard Nuclear Instrument Modules*. EG&G PARC provides the Model 4152, a six-wide bin equipped with the Model 7018 Power Supply, as a component of the 4100 system.

2.3C GROUNDING
The 4121's chassis is internally returned to the third wire or grounding pin of the standard bin's input power connector. For equipment and operator safety, this grounding pin should be returned to a good earth-ground connection via the ground wire of the power distribution system. This precaution serves to prevent shock hazards.

2.3D VENTILATION
The 4152 bin does not incorporate forced air ventilation.

2.4 FRONT PANEL
The 4121B is well designed from a human engineering point of view. With the exception of a single rear-panel switch, all of the controls are conveniently grouped at the front panel in functionally delineated areas. These controls and other front-panel features are discussed in the following paragraphs.

2.4A SIGNAL CHANNEL
The **SENSITIVITY** selector switch, **OFFSET** dial, and **OVLD** indicator are located in this section of the panel.

± FS Scale Sensitivity: This switch allows you to select any one of seven different full-scale sensitivities ranging from ±20 mV to ±2 V in 1-2-5 sequence. A full-scale synchronous signal will give ±10 V out (exponential averaging; fully converged). With linear averaging, a full-scale signal will give 10 V out if the number of samples taken equals 67% of the number selected by the **Samples Averaged** switch.

Offset: This ten-turn dial sets the level of dc offset from an internal source. It can be used to null offset on the applied signal or to compensate for stray internal drift and offset.

The range of the offset with respect to the input is ±10 times full scale up to a maximum of ±2 V. Maximum negative offset occurs with the dial fully counter clockwise and maximum positive offset when it is fully clockwise. Zero is at nominally five turns from either extreme. *Note that the 4121B must be triggering to see the effect of the offset setting at the output.*

Overload Indicator: This indicator will light for a minimum of 0.1 s any time the applied input exceeds ±2.2 V or the output exceeds ±10 V.

2.4B GATE WIDTH
The gate width (sample duration) equals the product of the **Range** switch setting times the **Multiplier** setting. The **Range** switch has settings from 1 ns to 30 μs. The range of the **Multiplier**, a one-turn dial, extends from X1 to X5. Thus the total gate-width setting range extends from 1 ns to 150 μs.

An analog of the gate pulse is provided at the **Gate Mon** connector, allowing you to use an oscilloscope to determine the precise duration and timing of the gate opening. The monitor pulse is 0.3 V (into 50 Ω to ground) and its width equals the gate width. There is, however, a small intrinsic delay between the actual gate opening and the monitor pulse. This delay will not exceed 5 ns when the **Gate Mon** output is loaded with 50 Ω.

2.4C OUTPUT
This area of the panel contains the **Samples Averaged** switch, the **Reset** pushbutton, and the two output BNC connectors, **AVG** and **LSO**.

Samples Averaged: This switch provides settings from 1 to 10,000 in 1-3-10 sequence. The significance of this parameter depends on whether the 4121B is doing exponential or linear averaging (averaging type is selected by internal jumper).

If exponential averaging, the setting defines the number of samples of which the output is the RC weighted average after nominally five times the selected number of samples have occurred. When the number of samples taken equals the switch setting, the output will be at 63% of its final value. The number of samples required to converge within 1% of the final value is five times the switch setting.

For example, if the **Samples Averaged** switch were set to **100** with a full-scale signal applied, the 4121B's output would reach 6.3 V after 100 samples and 10 V after 500 samples. At that point the Signal-to-Noise Improvement Ratio (SNIR) would be 10 ($100^{\frac{1}{2}} = 10$). If the number of samples taken is less than five times the setting, the SNIR will be less than $N^{\frac{1}{2}}$. Taking more than 5N samples provides no further improvement in SNIR.

If linear averaging, full-scale output occurs when the number of samples taken is 67% of the number selected by the Samples Averaged switch. Independent of the **Samples Averaged** switch setting, when linear averaging, the output is always the sum of the number of samples taken. The SNIR equals the square root of the number of samples taken.

Reset: Pressing this pushbutton resets the integrator, establishing zero output. You can also reset the integrator by applying a TTL logic 0 to the rear-panel RESET connector.

In linear averaging, all samples are summed so that, for a given input, the output level increases with each sample taken. This increase in output level doesn't stop until output overload occurs. The **Reset** function provides the only way of restoring zero output. You would normally do this at the end of each run.

In exponential averaging, the output approaches final convergence asymptotically. As a result, providing the input signal is less than full scale, output overload won't occur.

AVG Connector: The averager output is provided at this connector. Full-scale output is ±10 V and the output impedance is 50 Ω, with the constraint that the load impedance must be larger than 2 kΩ.

LSO Connector: The last (most recent) sample taken appears at this connector. Maximum output is ±10 V. The output impedance is 50 Ω, with the constraint that the load impedance must be larger than 2 kΩ.

This output defines the last (most recent) sample, that is, the average input level for the duration of the most recent gate opening. It is provided for both signal and baseline samples. It is not a function of the level at the **AVG** connector nor of the setting of the Samples Averaged switch.

In a 4100 system, the LSO Output of 4121B #1 is cabled to the Channel A input of the 4161A. Similarly, the LSO Output of 4121B #2 is cabled to the Channel B input of the 4161A.

2.4D INPUT

The two signal input connectors and the Coupling switch are located in this section of the panel. Although both inputs are active at all times, for best operation use one or the other, not both.

50 X Input: This input provides a 50 Ω input impedance and is always dc coupled.

1 MX Input:

This connector provides a 1 MΩ input impedance. It can be either ac or dc coupled as selected by the adjacent Coupling switch. **Keep this connector capped when not in use. It is always live.**

Coupling: This toggle switch allows you to select either ac or dc coupling for the 1 MΩ input. The 50 Ω input is always dc coupled.

2.4E GATE MONITOR

This section of the panel contains but a single BNC connector, **Gate Mon.** This output provides a marker pulse (0.3 V into 50 Ω) that defines the time position and duration of the gate opening. It lags the actual gate opening by no more than 5 ns.

2.4F TRIGGER

In addition to the Trigger Input connector, this section contains the Trigger Indicator, the Trigger Mode toggle switch, the Internal Oscillator frequency setting controls, and the Delay setting controls.

Trigger Input Connector: A sample is taken each time a trigger is applied to this connector (selector to ECL or TTL as appropriate). **Note:** When ECL is selected, the input impedance is 50 Ω to -2 V. When ECL is selected, the trigger source must develop ECL levels into 50 Ω returned to -2 V.

In a 4100 system, the A and B outputs of the 4144 connect to the TRIG IN inputs of 4421B #1 and 4421B #2 respectively. With this configuration, sample-delay is obtained via the 4144 by delaying the applied trigger. If the 4121B is not being operated as a 4100 system component, and there is no provision for generating delay ahead of the 4121B Trigger Inputs, delay can be introduced using the 4121B's internal delay capability.

Trigger Indicator: Each time the 4121B is triggered, either internally or by an externally derived trigger applied to the Trigger **IN** connector, this light flashes for at least 0.1 s. At rapid trigger rates, the light appears to glow steadily.

Mode Selector: This toggle switch determines whether the 4121B is configured for internal or external triggering, and, if external triggering, whether it will accept TTL or ECL pulses. Setting the switch to either **ECL** or **TTL** selects the **External** trigger mode. If the trigger source is TTL, set the switch to **TTL**. If the trigger source is ECL, set the switch to **ECL**. ECL triggering takes place on the **positive-going** edge of the applied trigger and TTL triggering takes place on the **negative-going** edge. An ECL trigger must be at least 4 ns wide and a TTL trigger must be at least 15 ns wide.

The maximum trigger rate in the External Trigger Mode is 100 kHz, unless baseline sampling is selected, in which case the maximum trigger rate is 20 kHz. Triggers applied while the 4121B is still executing its response to the previous trigger will be ignored. The **BUSY OUTLo** signal provided at the two rear-panel Interface connectors is asserted when processing of the sample initiated by the previous trigger is complete and the 4121B is trigger receptive again.

If the Trigger Mode toggle switch is set to **INT**, the 4121B is triggered by its internal oscillator. The oscillator's output also appears at the rear-panel **Int. Osc. Out** connector so that it is available for synchronizing peripheral equipment. The two **Internal** controls at the lower left corner of the front panel set the oscillator frequency.

Internal: The internal trigger mode frequency setting controls, both of which are set with a screwdriver, consist of a range switch and a vernier. The **Range** switch has six positions. Each position (other than **OFF**, in which the oscillator is turned off), defines a frequency range. The actual oscillator frequency within the selected range is set by the adjacent **MIN MAX** multiplier. The range for each position of the switch is as follows.

SETTING	NOMINAL RANGE
.50	.5 Hz to 5 Hz
5	5 Hz to 50 Hz
50	50 Hz to 500 Hz
500	500 Hz to 5 kHz
5000	5 kHz to 40 kHz

Delay: The Delay vernier and switch, active whether triggering externally or internally, set the delay between the trigger and the sample.

The **Range** switch has six positions, 3 ns, 10 ns, 30 ns, 100 ns, 300 ns, and 1000 ns. Each position corresponds to the delay obtained if the associated vernier is set to **MAX.** In the **MIN/EXT** position, the delay is the minimum possible for that range (see specifications).

> **Note:** The rear-panel **Delay Scan In** connector is always active and a voltage applied to this connector has the same effect as the front-panel Delay controls.

A potential applied to the rear-panel **Delay Scan In** connector provides a calibrated delay if the Delay Vernier is set to **MIN/EXT.** 0 V applied gives minimum (intrinsic) delay. +10 V gives a delay equal to the Delay Range setting. If a ramp voltage is applied, the delay linearly increases as the ramp advances from 0 V to +10 V. **Note:** If the vernier isn't set to **MIN/EXT,** the resulting delay won't be calibrated.

The intrinsic delay of nominally 20 ns when ECL triggering is always present and is in addition to the range selection. TTL triggering adds an additional 5 ns.

2.5 REAR PANEL

Besides the NIM power connector, rear panel features include the Sampling Mode toggle switch and six BNC connectors.

BNC CONNECTORS

1. **RESET AVG I/OLO:** TTL logic 0 applied to this connector resets the averager and initializes the logic. Alternatively, pressing the front-panel **Reset** pushbutton causes a TTL logic 0 at this connector. If the **Reset** connectors on two or more 4421B's are cabled together, pressing the **Reset** pushbutton on any one will reset them all.

   > **Note:** The reset must come from an open-collector source because the resets are tied together. Even if the resets from other modules aren't connected to the common reset line, an open-collector source is necessary to accommodate the front-panel Reset button.

2. **BUSY OUTLO:** When the 4121B is triggered, this line goes low. It remains low until the sampler is reset and the 4121B can be triggered again. When the 4121B is operated as part of the 4100 System, the **Busy** connectors are connected as follows.

FROM	TO
4121B #1 BUSY	4121B#2 BUSY
4121B #2 BUSY	4161A BUSY
4161A BUSY	4144 TRIG INH

3. **SAMPLE VALID OUTLO:** This TTL output goes low when the 4121B is triggered. It remains low until the resulting sample is valid. In a 4100 System, the **SAMPLE VALID** output of 4121B #1 connects to the A TRIG Input of the 4161B. Similarly, the **SAMPLE VALID** output of 4121B #2 connects to the B TRIG input of the 4161B.

4. **INT. OSC. OUT:** The output of the internal trigger

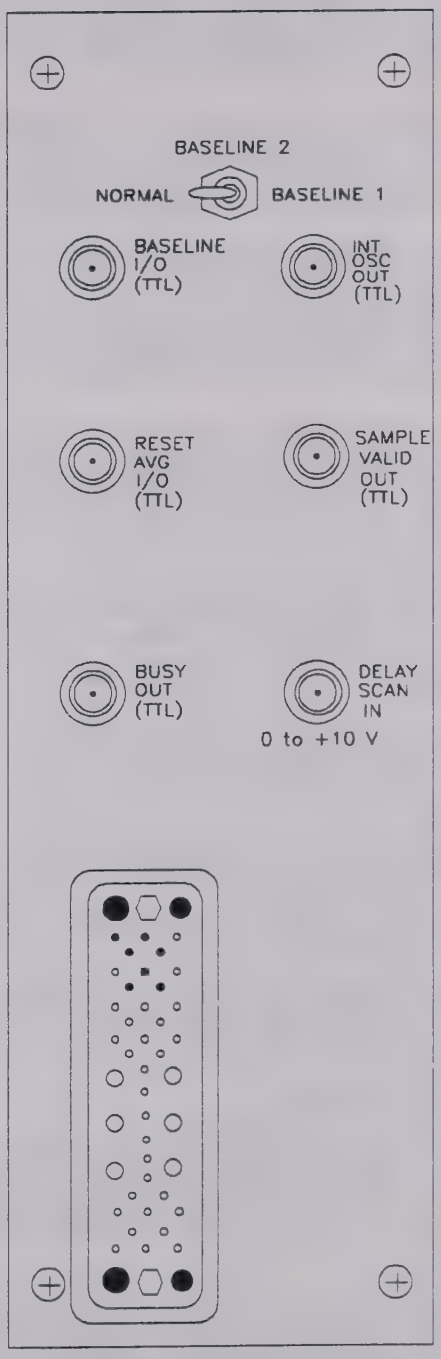


Figure 2-2. REAR PANEL

oscillator is provided at this connector. It is available any time the front-panel Trigger Range switch is set to any setting except **OFF**. It is not necessary that the Trigger Mode switch be set to **INT**. An internal jumper allows you trigger on the positive-going or negative-going edge of the oscillator output.

5. **DELAY SCAN IN (0 - +10 V):** Potential applied to this connector determines the delay if the front-panel Delay Vernier is set to **MIN/EXT**. Delay can be varied from minimum (intrinsic delay) with 0 applied (or open) to value selected by Delay switch (+10 V applied).

6. **BASELINE I/O:** This TTL I/O connector works in conjunction with the Baseline Sampling Mode switch as described in the following paragraphs.

SAMPLING MODE TOGGLE SWITCH
This switch has three positions, **Normal**, **Baseline 1**, and **Baseline 2**.

When it is set to **Normal**, samples are taken with respect to ground and every sample directly contributes to the averager output. This selection offers the advantages of fastest point acquisition and high trigger rates. The disadvantage of **Normal** is that signal baseline shifts can cause corresponding output level shifts and so act as a source of error.

In the other two positions, samples don't contribute directly to the averager output. Instead, two samples are taken, one of the signal and the other of the baseline, and the difference is transferred to the averager. This technique compensates for baseline drift. The maximum error due to baseline drift is reduced to that which can occur between the time the signal sample is taken and the time the baseline sample is taken. Disadvantages of baseline sampling are that twice as many triggers are required to make a measurement and that the maximum trigger rate is reduced from 100 kHz to 20 kHz.

Note that there is no provision in the 4121B for establishing a different value of delay for a baseline sample than for a signal sample. You have to do this externally by switching the trigger time between the baseline and signal delay values. The **Baseline I/O** function simplifies this task.

Baseline 1: In this mode, **Baseline I/O** functions as a TTL output that reverses state with each trigger. The reversal occurs synchronously with the negative-going edge of a TTL trigger and with the positive going edge of an ECL trigger. If the baseline output is high when the sample is taken, it is a signal sample. If it is low, a baseline sample is taken. After the baseline sample is taken, the difference between the baseline sample and the preceding signal sample is applied to the averager. Thus the averager output is incremented every other trigger. The state of the **Baseline I/O** output can be sensed by peripheral apparatus to switch the timing of the triggers so that baseline and signal samples are properly timed.

Baseline 2: In this mode, **Baseline I/O** functions as an **input** so that the experiment can directly control the sampling mode of the 4121B. When this input is high, the 4121B takes a signal sample. When this input is low, it takes a baseline sample and the difference between the baseline sample and the preceding signal sample is transferred to the averager.

NIM CONNECTOR
The power and ground connections are made via this connector, as follows.

PIN	USE
11	-6 V
17	-12 V
28	+24 V
29	-24 V
34	GROUND
16	+12 V
10	+6 V

2.6 USING THE 4121B

2.6A CONNECTIONS

In a typical experiment, you need to be concerned with connecting the input signal to one of the two **Input** connectors and a synchronous trigger to the **Trigger In** connector. Chances are you will want to monitor the **AVG** or **LSO** output as well with an oscilloscope. Alternatively, you may want to connect one of these outputs to the input of an Analog to Digital Converter (4161A) so that it can be digitized for computer processing. Also, you might want to monitor the **Gate Mon** output with an oscilloscope so that you can precisely determine the sample position. If you have need for the output of the 4121B's internal trigger oscillator, its output, a TTL square wave at the frequency set by the front-panel Internal Trigger controls, is available at the rear panel **INT OSC OUT** connector. If you elect to use baseline sampling, you will have to consider the rear-panel **Baseline I/O** connector too. In the **Baseline 1** mode, this connector functions as an **output**. In the **Baseline 2** mode, it is an input.

If the 4121B is operated as a component of the 4100 System, it will be connected as follows. Two 4121B's are assumed, the first designated **A** and the second **B**.

UNIT A
Signal Input: signal to be measured
Trig In: to 4144 **A** Output
LSO Out: to 4161A **Chn A** ADC Input
Sample Valid Out: 4161A **Chn A** ADC Trig Input
Busy: All system Busy's tied together

UNIT B

Signal Input: signal to be measured
Trig In: to 4144 **B** Output
LSO Out: to 4161A **Chn B** ADC Input
Sample Valid Out: 4161A **Chn B** ADC Trig Input
Busy: All system Busy's tied together

2.6B SIGNAL SENSITIVITY

The sensitivity should be set to give as near full-scale output as possible without overload. Since the maximum input that can be applied without overload is ±2 V for any sensitivity, if the signal is very noisy, it may be necessary to attenuate it ahead of the 4121B to avoid input overload. Offsets, both external and input, can be nulled using the ten-turn **OFFSET** control. Note, however, the unit must be triggering to see the effect of the adjustment.

2.6C GATE WIDTH

The gate width sets the sample duration and hence the resolution. The narrower the gate, the finer the signal detail you will be able to recover. Delay jitter can be a significant resolution-limiting factor. Relatively speaking, its impact is greater with a narrow gate width than with a wide one. (Delay jitter in the 4100 system is determined by the 4144 Digital Delay Generator; see specifications in Chapter 3 for details.)

2.6D OUTPUT

The **Samples Averaged** switch directly sets the attainable SNIR when exponential averaging. The higher the setting, the better the SNIR, but at the expense of increased experiment time. Unfortunately, whereas experiment time varies directly with the setting, SNIR varies with the square root of the setting, tempering the practicality of always operating with a high samples averaged setting.

If linear averaging, the **Samples Averaged** switch simply sets the number of full-scale input samples (67% of setting) needed to develop a full-scale output. It has no effect on SNIR.

Keep in mind that baseline sampling doubles the number of triggers required. This is because the difference between sample pairs is averaged and not every sample.

The averager output is provided at the **AVG** connector and the most recent sample level at the **LSO** connector. The **RESET** button can be used to reset (zero) the averager at any time.

In an experiment where it is necessary to precisely know when the output has settled so that it can be processed, the rear-panel **SV** (SAMPLE VALID) output provides the required information. This output goes to logic 0 when the 4121B is triggered. It remains at logic 0 until any output change from the resulting sample has had time to stabilize, at which point it goes to logic 1. This positive-going transition could be used to trigger the A/D converter.

2.6E INPUT

If you've selected the 50 Ω input, there are no further input decisions to be made. On the other hand, if the input signal is connected to the 1 MΩ input, you must be decide between ac or dc coupling.

2.6F TRIGGERING

To trigger internally select **INT** and then the desired frequency. To trigger externally, select either **ECL** or **TTL**, whichever applies, and connect the trigger source to the **TRIGGER IN** connector.

The delay between the trigger and the sample is set using the **DELAY** controls. Since the inherent delay is fairly substantial compared to the shorter delay ranges, it's probably a good idea to monitor the signal at the **GATE MON** connector to ascertain the precise gate position.

Each time the 4121B is triggered, it ignores subsequent triggers until all actions initiated by the previous trigger have been completed. The rear-panel BUSY OUT signal delineates the interval over which triggers are ignored. This open-collector output goes to TTL logic 0 as soon as the 4121B is triggered. When the cycle thus initiated is complete, BUSY OUT goes to logic 1, indicating that the 4121B can be triggered again.

2.6G BUSY OUT VS SAMPLE VALID

Transitions in these two outputs define two different points in the cycle that follows each trigger. When the 4121B is triggered, both **BUSY** and **SAMPLE VALID** go to logic 0. Following the appropriate delay, the sample is taken. At some point after that, the **AVG** and **LSO** settle, that is, they become valid, and **SV** (SAMPLE VALID) then goes positive. This positive-going transition can be used by equipment following the 4121B as a marker indicating that it is all right to process the 4121B outputs.

Assertion of **SV**, however, does not mark the end of the cycle. The sampler circuits are automatically reset at the end of every cycle and this takes some additional time. When the reset has been accomplished, BUSY OUT goes to logic 1, indicating that all activity has ended and that the 4121B can be triggered again. Thus this output becomes important in an experiment where the highest possible trigger rate must be maintained.

2.7 JUMPERS

Some operating configurations are established by jumpers on the 4121B Signal Board. Figure 2-3 is a Signal board layout diagram that shows the location and function of the user accessible jumpers.

Averager Mode: Jumper J9 selects the averaging mode, either exponential or linear. J9 is a two-width jumper that mounts on two of the three in-line pins located just below the end of the Samples Averaged switch. If the jumper is in the upper position (center pin and top pin jumpered together), **EXPONENTIAL** is selected. In the

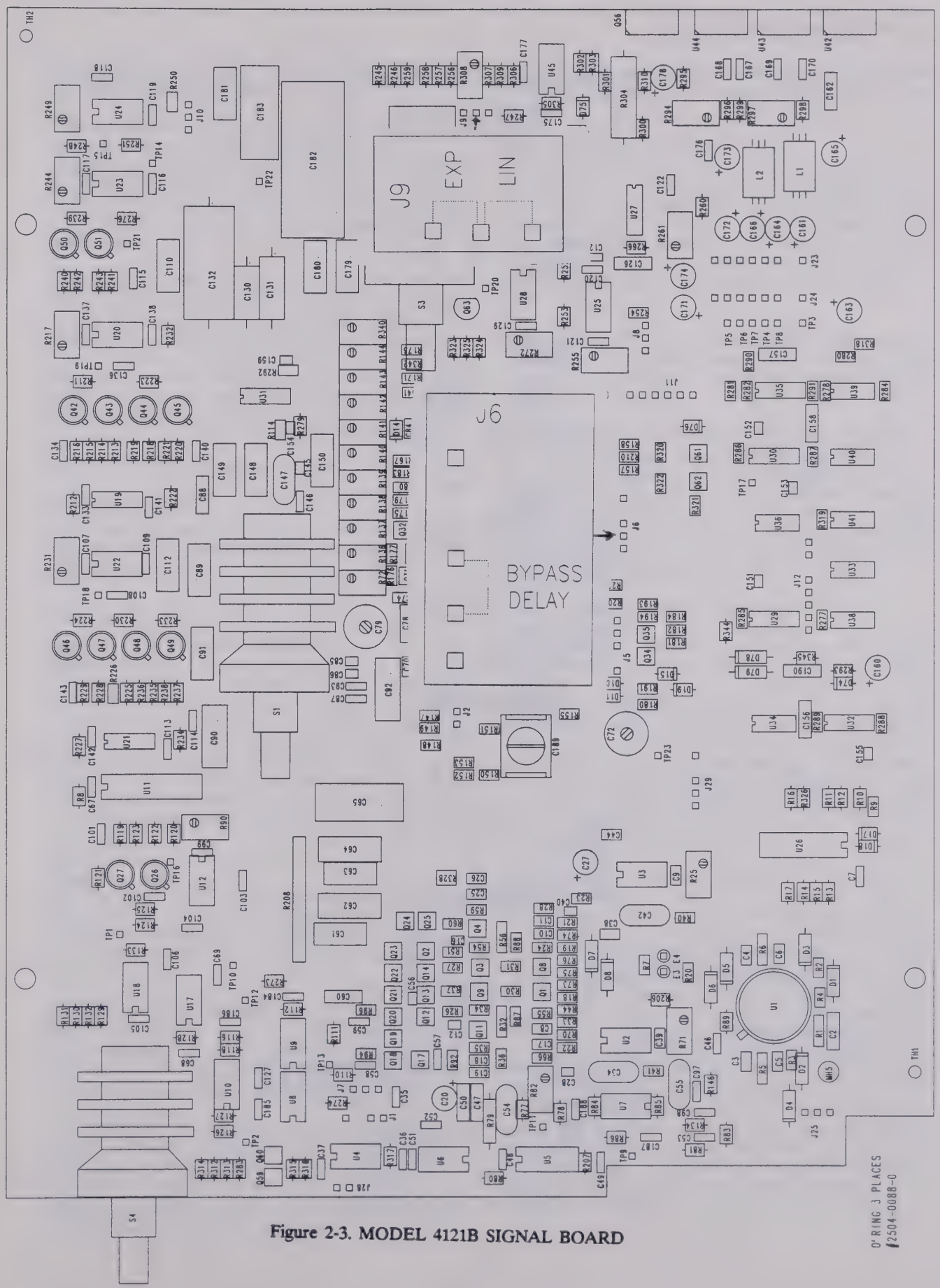


Figure 2-3. MODEL 4121B SIGNAL BOARD

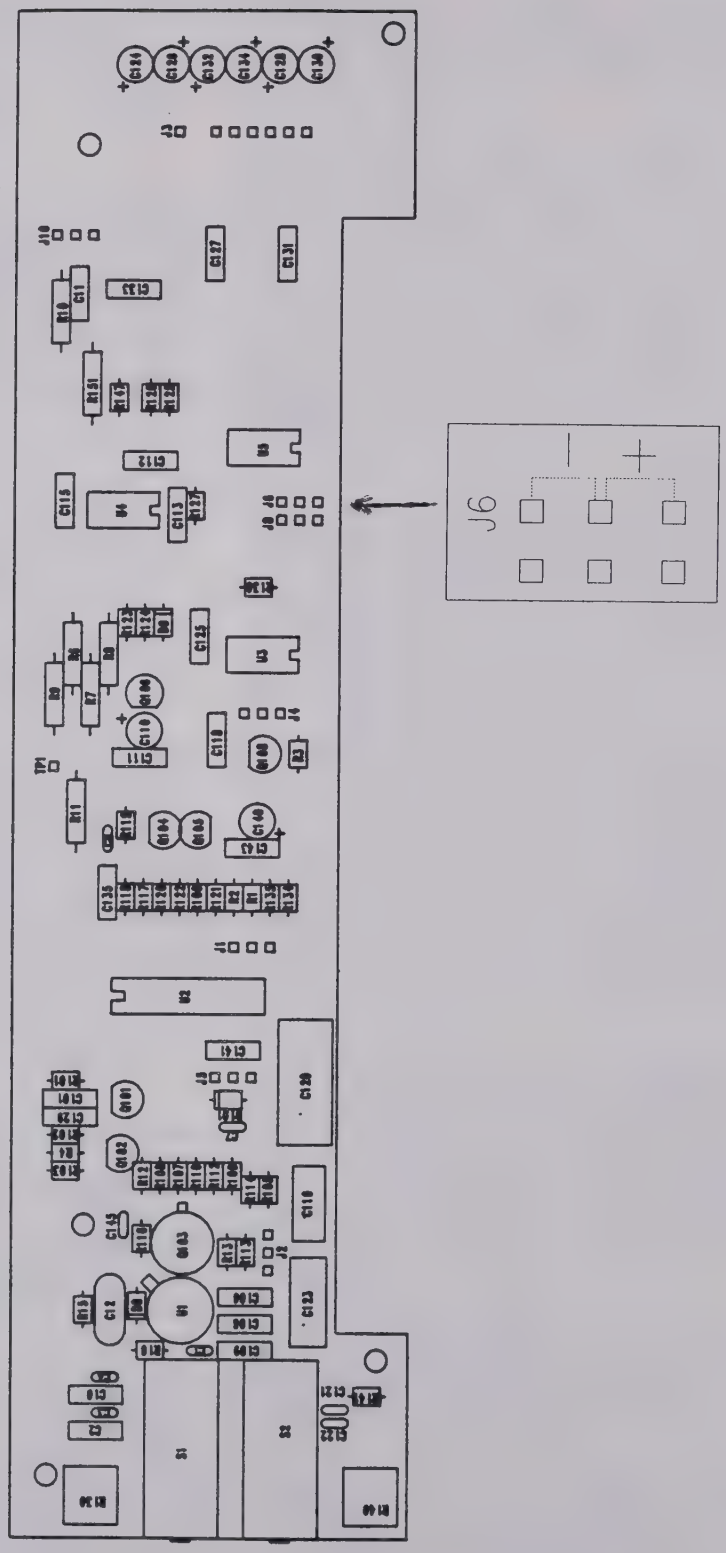


Figure 2-4. MODEL 4121B DELAY BOARD

lower position (center pin and bottom pin jumpered together), **LINEAR** is selected.

Delay Bypass: Jumper J6 allows the delay function to be bypassed to obtain the minimum possible delay between triggering and sample acquisition. Figure 2-3 illustrates the J6 Bypass Selection. Normal "Delay In" operation occurs with no jumper block installed. To bypass the delay function, jumper pins 2 and 3 (counting left to right) together.

Internal Osc. Polarity: Jumper J6 on the DELAY BOARD determines whether triggering will take place on the positive going or negative going edge of the trigger oscillator output. The position of this jumper is shown in Figure 2-4. As indicated, there are three pins. When pin 2 (center pin) is jumpered to pin 1 (pin closest to edge of board), triggering occurs on the positive-going edge. Jumpering pin 2 to pin 3 selects triggering on the negative-going edge.

2.8 CIRCUIT DESCRIPTIONS

2.8A INTRODUCTION

The following pages provide a moderately detailed description of the Model 4121B circuits. Schematics are located in Appendix H.

2.8B SIGNAL INPUT

2.8B1 HIGH IMPEDANCE BUFFER

(Signal Bd. sh 1)

The signal applied to the **INPUT** BNC connector is either ac or dc coupled by the Coupling switch (S2-A) to the input of U1. The input impedance as established by R4 is 1 MΩ. Diode clamping networks to 5 V through 10 kΩ provide overvoltage protection. The output of U1, a 100 MHz unity gain buffer, drives R18 and R76. These resistors in turn drive the emitters of Q1 and Q8, the common-base input stage.

2.8B2 50 OHM DC INPUT

(Signal Bd. sh 1)

The front-panel 50 Ω dc input directly drives R73, R75 and R7, giving a 50 Ω input impedance. R73 and R76 connect to the emitters of Q1 and Q8, respectively, the complementary input stage. The maximum input is +5 V.

2.8C SAMPLING AMPLIFIER

(Signal Bd. sh 1)

The collectors of Q1 and Q8 provide complementary signal currents to the emitters of Q9 and Q3, the common base stage that drives the Sampler Bridge.

The emitter of Q1 is monitored by an operational amplifier (U2 pin 2) via R70, C28, and R22. The output of U2 (pin 6) drives the base of Q1 via R66, C17, C8, and R55, to maintain zero emitter voltage at Q1. Similarly, the emitter of Q8 is monitored by operational amplifier (U3 pin 2) via R21, C40, and R23. The output of U3 (pin 6) feeds back to drive the base of Q8 via R28, C11, C10, and R24 to maintain zero emitter voltage at Q8.

2.8D SAMPLING BRIDGE

(Signal Bd. sh 1)

Q9 and Q3 drive the Sampling Bridge, which consists of a PNP differential pair, Q12 and Q13, and an NPN differential pair, Q14 and Q2. When not sampling, the bias current flows between the PNP and the NPN pair. Any net current (signal or bias error) in the NPN or PNP collector is sensed by the virtual ground at pin 2 of U6.

When a signal is sampled, the Sampling Bridge switches the current to its complementary half (Q13 and Q14). This signal current is then integrated onto the selected sampling capacitor.

Q15 and Q16 provide a 600 mV differential drive to the PNP differential pair with an average dc value of 3 V. Q65 and Q64 provide a 600 mV differential drive to the NPN drive, with a -3 V average dc value, and, together with Q15 and Q16, determine the gate width. Q15's collector drives R135 and R39, and Q16's collector drives R327 and R38. Q65's collector drives R337 and R53. Q64's collector drives R38 and R52.

2.8E SAMPLING CAPACITORS

(Signal Bd. sh 2)

The Gate Width switch (S1), in addition to controlling the gate width as described further on, selects a different sampling capacitor in each switch position. Each capacitor is paralleled by a 1 MΩ resistor to bias the series FET source at zero dc voltage for gating control. Each capacitor (C56 through C65) is in series with a FET (Q17 through Q24) controlled by the front-panel Gate Width switch. Any one of the selected sampling capacitors is turned on by the Gate Width switch, pulling the appropriate gate to +12 V. The series FET's have an on resistance of 50 Ω or less.

2.8F BRIDGE BALANCE CIRCUIT

(Signal Bd. sh 1)

Any current unbalance in the Sampling Bridge will cause an input at pin 2 of U6. The output of U6 drives pin 3 of U5, a current buffer amplifier. It's output (pin 8) is a voltage proportional to the unbalance current times the 464 Ω R77 feedback resistor. The output of U5 is summed into pin 2 of Integrator U7 via R78 and R82. The signal current is always summed into pin 2 of U7 via any one of the three networks, the High-Impedance Buffer Output driving R89, C46, R83, C53, and R81, or the 50 Ω Input driving R206, C97, R146, C98, R134, or the DC Offset Input driving R207. Any differences arising due to temperature and beta drifts will cause a voltage at U7 pin 1, which is inverted by U7, pin 7, and fed back through resistors R87 to the emitter of Q9 and R88 to the emitter of Q3 to balance the bridge.

2.8G SAMPLER RESET
(Signal Bd, sh 1)
MOSFET Q10 is immediately turned off (Q6 & Q7 drive Q10's gate to -2.5 V) when a trigger is received, allowing the sampling capacitor to take on a charge proportional to the signal. It is then reset (Q6 and Q7 drive Q10's gate to 8 V), but only after the post samplers, i.e., the LSO Sampler and the Averaging Sampler have been updated.

The collector of Q6 drives the gate while the complementary drive (Q7) drives C45 to couple a complementary signal for capacitive neutralization.

The base drives for Q6 and Q7 are the attenuated (R63, R62, R65 & R67) outputs of ECL gate U14 (pins 2 and 3).

2.8H DC OFFSET
(Signal Bd. sh 1)
The dc offset generated by the OFFSET potentiometer is applied to buffer amplifier U4, which is current buffered by Q59 and Q60. The common emitters of Q59 and Q60, via jumper J28 drive R44 and R19. These resistors drive the emitters of Q1 and Q8. This has the effect of subtracting dc current from the signal current at the emitters Q1 and Q8. The range of the offset is a maximum of ten times full scale. Feedback resistors (R283, R312, R313, and R314) control the range for any setting of the Sensitivity switch by controlling the feedback to U4.

2.8I Q1-Q8 DC ZERO
(Signal Bd. sh 1)
Operational amplifier U2 monitors the emitter of Q1. U2's output in turn is fed back to the base of Q1 to keep the emitter potential of Q1 at 0 V. Similarly, U3 monitors the emitter of Q8 and its output is fed back to the base of Q8 to maintain Q8's emitter potential at 0 V.

2.8J DC AMPLIFIERS
(Signal Bd. sh 2)
The differential amplifier comprising U8, U9, and U10 amplifies the voltage across the selected sampling capacitor. The gain varies over a range of X2 to X10 and is adjusted by the front-panel gate vernier potentiometer. The output (pin 6 of U10) is provided at test point TP2.

U10's output is routed through U17 and U18, which provide additional gain. Sections 4B and 4C of the Sensitivity switch set the gain of these stages. The output of U18, the final dc signal ahead of the Output Sample and Hold circuits, can be monitored at TP1.

2.8K DC STABILIZATION
(Signal Bd. sh 2)
A sample and hold circuit comprising U12, Q26 and Q27, also monitors the output of U17. Its purpose is to maintain TP1 at 0 V. The potential at TP1 is sampled after the sampling capacitors have been reset. If the voltage there is other than zero, the output of integrator U12, fed back to pin 6 of U9, maintains 0 V at TP1.

2.8L LSO SAMPLER
(Signal Bd. sh 4)
The LSO sampler consists of U20 working in conjunction with FET switches Q42 and Q43. After each sample, this circuit samples and stores the dc amplified voltage on the selected sampling capacitor and provides it through further amplification to the LSO OUTPUT. Capacitor C136 is the sampler's storage capacitor. Another capacitor, C110, extends the holding capability of the sampler when the duty cycle is low.

The LSO Sample and Hold gating time, during which the LSO Sampler sees the voltage on the sampling capacitor, is about 2.5 ls as set by U19-A, a one-shot pulse generator. The pulse out of U19 is synchronous with the trigger but delayed by the time needed to take the signal sample and for the voltage at TP1 to settle. U19-B turns Q44 on, thus bringing C110 into the circuit, if the trigger frequency drops below nominally 500 Hz. The output of the LSO sampler is amplified by U24 (X5) and routed to the front-panel BNC connector through a 50 Ω resistor.

Diodes D17 and D18 monitor the LSO sampler output for an overload condition. Should an overload potential occur, it will be detected by D17 and D18, which drive the Overload Circuit to light the front-panel OVLD detector.

2.8M BASELINE SUBTRACT BUFFER
(Signal Bd. sh 4)
If the sample at the output of the LSO Sampler is a baseline sample, it is routed through FET switch Q51 to inverter U23, so that it can be combined with the signal sample at the input of the Averager. Because of the inversion, the Averager sees the difference of the signal and baseline samples.

2.8N SIGNAL OUTPUT SAMPLER
(Signal Bd. sh 4)
U22, the Signal Output Sampler, is quite similar to the previously described LSO Sampler. The principal difference is that, during baseline subtract mode operation, the effective trigger rate of the Signal Sampler is one half the input trigger rate, and the sampler is used to store the signal sample only, thus allowing the difference between the LSO Sampler output and the Signal Sampler output to be applied to the averager. Capacitor C108 is the sampler's storage capacitor. Capacitor C112 extends the holding capability of the sampler when the duty cycle is low. U21-A drives the Sample and Hold. U21-B switches C112 into the circuit if the trigger frequency falls below approximately 500 Hz.

2.8O AVERAGER
(Signal Bd. sh 4)
U23 pin 6, which is the inverted output of the LSO

Sampler, is turned on in baseline-subtract mode only. Together with U22, pin 6, the output of the Signal Sampler, it is summed into the averager through resistors R245 and R246. The junction of these resistors drives the averaging resistors (R256, R257, R258, R259) through the Samples Averaged switch to the summing junction of Q28 via FET switches Q52 and Q53.

Similarly, the averaging capacitors, C130, C131, & C132, are connected to the summing junction of U28 by the Samples Averaged switch.

Q55 and Q57, which are connected in series across the averager, reset the averager. The junction of Q55 and Q57 is grounded through R311. The purpose of this series connection is to lower the voltage drop across Q57 in the off condition to minimize leakage current into the averaging capacitors. The averager operates by allowing signal currents to be switched into the summing junction of U28 through Q53 for a period of approximately 5 ls as determined by one shot U27. Q54 and Q58, driven by U28, switch in a larger capacitor during low duty operation to reduce the droop rate. The output of the averager is amplified by non-inverting gain-of-five stage U25 and fed to the AVG OUT BNC connector through R54.

U28 pin 6 feeds back via R247 in series with J9, the Exp/Lin Jumper Selector, to the junction of R245 and R246. The purpose of the Exp/Lin switch is to select U28's averaging mode. Installing the jumper establishes the exponential averaging mode by connecting R247 in the feedback. Linear averaging occurs when the jumper is in the holding position.

2.8P GATE WIDTH GENERATOR
(Signal Bd. sh 3)
Current source Q31 supplies current to the capacitor selected by the front panel Gate Width switch, developing a linear voltage ramp on the capacitor. The front-panel Gate Width switch sets the emitter resistance and hence the ramping current provided by Q31. The reference base voltage for Q31 is provided by a temperature-compensating network (R175, R179, Q32, R177, C80, & R178).

The ramping current is available to the selected capacitor whenever Q29, controlled by flip-flop U14-B, is switched off. Quiescently, Q29 conducts and the selected capacitor is discharged to the preselected clamping level set by D13, a high-speed Schottky diode. This clamping level is temperature compensated by operational amplifier U16 which incorporates a feedback network containing D14, a diode of the same type as D13 and with the same bias current.

When U14 is triggered, Q29 switches off and the current supplied by Q31 to the capacitor develops a positive going ramp that is monitored by pin 2 of comparitor U15. At the same time, U14 pin 15 is applied to U13 (pin 24), which, through further logic buffering, generates the complementary gate drives to the sampler at the outputs of U13, pins 13 and 14, and U13, pins 5 and 6.

The gate-width vernier voltage from the front panel VERNIER potentiometer acts as the other input to U15 (pin 3). When the ramp voltage at pin 2 crosses over the potentiometer voltage at pin 3, U15's non-inverting output goes high and resets U14. At the same time, U15's output switches the sampler gate drives off. Q29 switches back into conduction to discharge the capacitor and restore the quiescent state. Gate U13-D produces the gate monitor output.

2.8Q TRIGGER INPUT
(Signal Bd. sh 3)
The trigger input is either TTL, internal, or ECL. In the ECL mode, diodes D9 and D11 provide over-voltage protection. The input ECL signal drives an over-voltage circuit whose output in turn drives pin 17 of U13, an ECL OR gate. The other input to this OR gate, pin 18, is the signal source when triggering from a TTL source. The TTL signal drives differential amplifier Q34-Q35. The output of this circuit is the appropriate ECL translated signal applied to pin 18 of gate U13. Pin 16, the non-inverting output of U13, goes to J6 pin 2, which is a strip of berg connectors. A jumper at J6 is used to apply the trigger to the Delay board or to bypass it, as desired. The delayed trigger appears at pin 3 of J67 and is applied to pin 6 of U14, the Trigger flip flop. This circuit shuts off the FET reset circuit and triggers U14-B (pin 11), the gate-width generator control flip-flop previously described.

In addition, differential pair Q61-Q62 generates a digital RUN signal. This signal immediately feeds back to pull the base of Q41 and Q36 to a logic low and initiate a trigger holdoff. When all of the samplers have completed their cycles, the system initiates a calibrate cycle by pulsing the base of Q38 to a logic low, thus presetting U14, which in turn resets the FET across the sampling capacitor. Q37's base is pulsed to logic low to start the dc stabilization cycle. When the stabilization cycle is complete, the base of Q38 is pulsed again to preset U14. The bases of Q41 and Q36 are set to a logic high to accept the next trigger.

2.8R FRONT PANEL GATE MONITOR
(Signal Bd. sh 3)
Pin 7 U13 is an ECL level signal whose width is equal to that of the sampling gate pulse. It is dc level shifted to generate a 0 to +0.3 V pulse into 50 Ω at the front-panel Gate Monitor BNC connector that indicates the position and width of the sample.

2.8S DELAY BOARD
Each trigger at J6 pin 2 of the main board is routed to J1 pin 2 of the Delay board. This signal triggers ECL trigger flip flop U2-A. Pins 2 and 3 of this flip flop drive differential amplifier Q103 via attenuator network R107, R108, R114, & R113. In response, Q103's current is diverted from its clamping diode, D8, allowing the

constant current source generated by Q102 to linearly charge the capacitor selected by the Trigger Delay Range switch (S1). The resulting voltage ramp is applied to one input of comparator U1 at pin 2. The voltage from the Delay Vernier potentiometer, R130, is applied to the comparator's pin 3 input. When the ramp voltage at pin 2 crosses over the pin 3 potential, the output of the comparator (pin 7) goes high, switching U2, the trigger flip flop, and discharging the ramping capacitor. U1 pin 7 is also fed to J2 pin 2, and from there to J6 pin 3 of the main board to start the sampling process.

The input trigger to the Delay Board also triggers U2-B, an ECL flip flop. The complementary outputs of this flip flop drive Q104 and Q105, a differential current source. The sole function of this circuit is to maintain a net current of zero out of the emitter of Q106, to maintain a constant reference voltage for the ramp generator.

A voltage (range 0 to +10 V) applied to the rear-panel Delay Scan BNC is applied via R110 to R8 and R9, which are used to develop the trigger delay voltage across the Delay Vernier potentiometer. The specified Delay Range for a voltage applied to the Delay Scan connector is obtained when the Delay Vernier potentiometer is set to MIN.

2.8T OSCILLATOR
(Delay Bd)
U3 is an oscillator whose frequency is controlled by capacitors selected by Switch S2 and Frequency Adjust vernier potentiometer, R140. The output of this oscillator is buffered by U5 to generate the complementary TTL 50 Ω oscillator output signal provided at the rear-panel INT OSC OUT connector. A jumper (J6) selects the sense of the TTL Oscillator output.

2.8U LOGIC

2.8U1 LSO GATING LOGIC
(Signal Bd. sh 5)
U30-A, a 2.5 ls One Shot, acts to delay the LSO sampler, allowing the amplifier time to settle. The output of this single shot, pin 4, is gated to allow only a single trigger to the LSO Sampler because an internal second trigger is generated to measure the offset generated by the Reset circuit. The output of the LSO Single Shot drives another single shot, U31-A, whose delay is controlled by the Gate Width switch. This delay allows the proper amount of time for discharging the sampling capacitor. The output of U31-A (pin 4) is then used to strobe a 100 ns single shot, U31-B. This starts the calibration cycle.

2.8U2 CALIBRATION CYCLE
(Signal Bd. sh 5)
The output of U31-A pin 4 also simultaneously triggers U30-B at pin 10. This one shot is used as a trigger inhibit to the LSO Sampler.

Upon completion of the calibration, U29-A (pin 2), a 100 ns single shot, is used to reset the logic that sets up for the next trigger. The output (pin 4) of U29-A clocks pin 11 of U34-B to remove the trigger holdoff. When U34 is clocked, pin 8 goes to logic 1, which acts through TTL to ECL level translator, Q36, to release the clock input to U14-A, the ECL Start flip-flop.

2.8U3 START LOGIC
(Signal Bd. sh 5)
Upon receiving an accepted trigger, U34-A pin 5 is preset to logic 1, which acts through buffer U39 (pin 6) to drive SAMPLE VALID to active low. This active low indicates the beginning of an acquisition cycle. This accepted trigger also presets U34 pin 8 to logic low, preventing any more triggers from being accepted. The accepted trigger also drives retriggerable one shot, U35 (pin 2), which supplies current to the front-panel TRIGGERED LED.

2.8U4 RESET LOGIC
(Signal Bd. sh5)
A rear-panel reset can be generated via an open-collector TTL logic low applied to the rear-panel RESET BNC connector. **Note:** *The reset must come from an open-collector source because the resets are tied together. Even if the resets from other modules aren't connected to the common reset line, an open-collector source is necessary to accommodate the front-panel Reset button.*

Internally, the duration of the reset command is the longer of: the width of the applied reset logic low or the output of single shot U32-A pin 4. The output of U32-A (pin 4) drives a three input NAND gate (U38-B) to combine the front-panel Manual Reset pushbutton circuit and the power-up reset logic. Pin 6 of this NAND gate is then used throughout to generate the averager reset logic and its complement through pin 8.

U38-B pin 6 also drives pins 1 and 2 of U39, an open-collector TTL gate that provides the rear-panel BUSY signal.

2.8U5 NORMAL vs BASELINE LOGIC
(Signal Bd. sh 5)
In the NORMAL sampling mode, the preset input of U33-A (pin 4) is grounded, disallowing baseline-subtraction operation. In the BASELINE 2 mode, the rear-panel BASELINE I/O connector acts as an input to determine whether the sample taken with the next trigger will be a signal sample or baseline sample. If the applied level is low, a baseline sample will be taken on the next trigger. If high, it will be a signal sample. **Note:** The signal at the LSO OUTPUT is never the difference but either the baseline or signal sample as appropriate. In the BASELINE 1 mode, U33-A toggles on alternate triggers to alternatively take a signal sample and then a baseline sample. The rear-panel BASELINE I/O connector acts as an output, toggling between TTL low (baseline) and TTL high (signal). Pin 6 of U33-A drives pin 9 of U40 to control the averager single shot.

Chapter 3
Model 4144

1.1 INTRODUCTION

The Model 4144 two-channel Digital Delay Generator offers unparalleled timing flexibility in a single unit. Two delays adjustable from 0-10 ms in 10 ps steps each culminate in an output pulse of width adjustable by preset potentiometers from 20 ns to 20 μs. The delays can be either fixed at separately determined values or scanned together. The unique Interruptible Ramp technique keeps jitter below 50 ps rms without calibration adjustments throughout the operating temperature range for an indefinite period. The availability of +5 V and +10 V outputs from low impedance sources allow 50 Ω, or higher or lower impedances, to be driven. Capacitive, inductive, or resistive loads can all be accommodated. The output circuits are automatically protected against overload.

FEATURES

- Two delay outputs
- Two Variable Widths & Two Gates
- Internal, External, Single Shot, Burst Mode & Scan Mode Triggers
- Trigger Rate to 2 MHz
- <50 ps rms jitter with no setup required
- Outputs: +5 V & +10 V from low impedance & Negative NIM
- GPIB Interface

APPLICATIONS

- Pulsed Laser Timing
- Radar Simulation
- Chemical Kinetics
- Sonar
- Fiber Optics Testing
- Shock-wave Physics
- Loran
- Time-of-Flight Measurements
- Jitter reduction in Oscilloscope and Boxcar Averager Trigger Delays

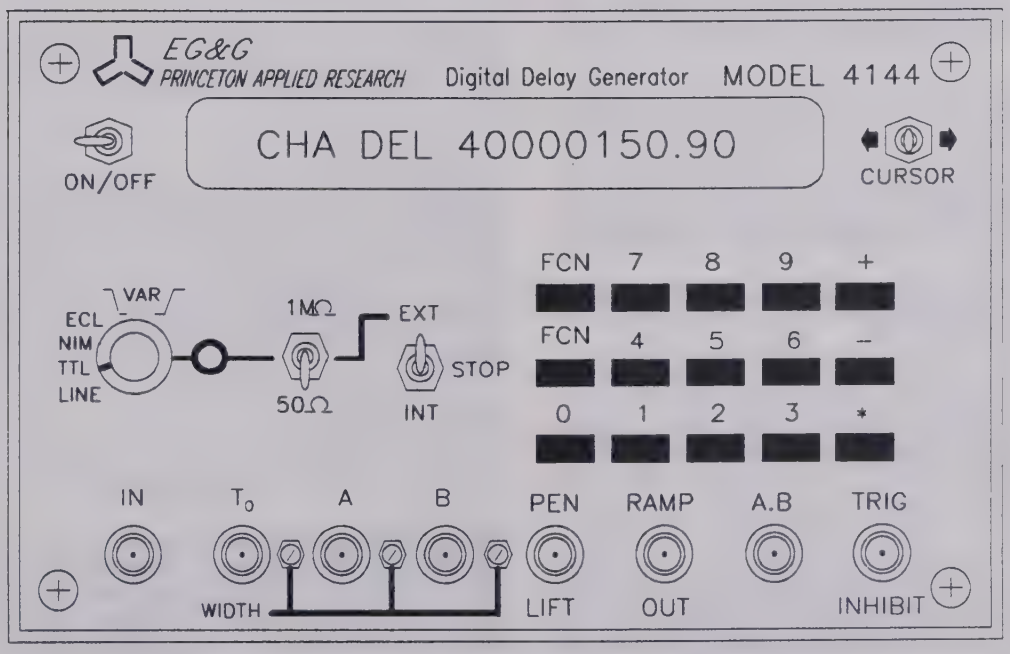


Figure 3-1. MODEL 4144 DIGITAL DELAY GENERATOR

3.2 SPECIFICATIONS

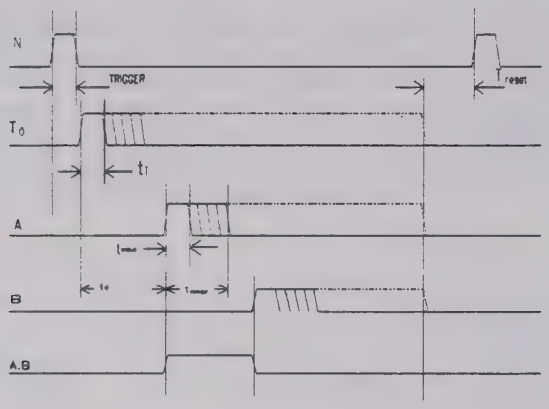


Figure 3-2. TIMING DIAGRAM

OUTPUTS:

T_o: Time Zero output. This is a variable-width (but not variable delay) output. The insertion delay (time between leading edge of trigger and leading edge of this pulse) is 35 ns max.

A and B: Variable-width, variable-delay outputs. The width of the A and B outputs is variable from 0.02 μs to 20 μs.

> **Note:** A and B may be in any order. The channels are identical in operation except that T_o and B cannot be longer than 200 ns + A when A > 100 ns, or longer than 300 ns + A when A < 100 ns. By making A the widest output, this constraint will never be a problem.

T_o, A, and B PULSE LEVELS: +5 V, +10 V or NIM (-16 mA into 50 Ω).

> **Note:** The 5 and 10 V outputs are low impedance and will drive low or high impedance loads including large capacitance loads. Risetime into 50 Ω load is typically 3 ns to +5 V, 5 ns to +10 V, and 1.5 ns to -16 mA. All outputs are short-circuit protected.

A.B: A composite output that starts at the A Output leading edge and ends at the B Output leading edge. The A.B function therefore provides a pulse with very low jitter leading and trailing edges and a width equal to the difference between the A Delay and B Delay settings and completely independent of the A and B width settings. Levels are 0 V & +5 V and it will drive 50 Ω or a high impedance. Either sense (up on A & down on B, or down on A and up on B) can be selected. This output is not active in scanned operation.

RAMP OUT: 0 to +10 V stepped ramp provided in scanned operation. It will drive a 1000 Ω load.

PEN LIFT: TTL output that goes to logic 1 at end of scan and that remains at logic 0 during a scan.

DELAY RANGE:
0 to 10 ms

DELAY ACCURACY:
Delay x timebase stability ±0.3 ns

INSERTION DELAY:
Maximum 35 ns

TIMEBASE STABILITY:
±20 ppm (0-50°C)

JITTER (rms):
Between trigger or any output and the successive output < 50 ps + 1 x 10^{-8} of setting.

TRIGGER MODES:

External: 0-2 MHz. Threshold variable ±2 V. Preset TTL, NIM, or ECL, switch selectible. Slope positive or negative. Input impedance 1 MΩ in parallel with 30 pF or 50 Ω. Minimum trigger width is 10 ns. Input protected to ±25 V dc and ±100 V at 10% duty cycle.

Internal: Single shot or .001 Hz to 1.000 MHz (4 digit resolution from 1 Hz to 1 MHz; .001 Hz resolution below 1 Hz). Accuracy same as timebase.

TRIGGER RATE:
External or internal trigger rate should be less than 1/(longest delay + 350 ns) for delays up to 80 μs and less than 1/(longest delay + 500 μs) for delays longer than 80 μs.

TRIGGER INHIBIT: Front-panel TTL input allows trigger and scan inhibiting for duration of logic 0 at this connector.

BURST MODE:
1 to 50,000 pulses using external or internal triggers at up to 20 kHz rate.

SCAN MODE:
An initial delay, delay step size, number of steps per scan, and number of triggers per step can be specified. 1 to 49,999 external or internal triggers per step may be used in scan mode at up to 20 kHz rate and up to 3999 delay steps are permitted.

SCAN INHIBIT:
Active low TTL rear-panel input inhibits outputs after scan ends. Must go high for 100 ns minimum to restart scan. No effect except in scan mode.

DISPLAY:
5x7 Dot Matrix alphanumeric vacuum fluorescent.

PARAMETER SETTINGS SAVE/RECALL:
All settings are automatically stored when power is removed and restored when power is reapplied. Up to 30 additional complete settings may be later recall.

IEEE-488 (GPIB) INTERFACE:
Allows delays, internal rate generator, burst and scan mode parameters to be controlled remotely.

GPIB Interface Functions Supported:

SH0 (Source Handshake)
AH1 (Acceptor Handshake)
T0 (Talker)
L2 (Listener)
SR0 (Service Request)
RL2 (Remote-Local)
PP0 (Parallel Poll)
DC1 (Device Clear)
DT1 (Device Trigger)
C0 (Controller)
E0

DIMENSIONS:
8.25" wide by 4.75" high x 11" deep (20.96 cm x 12.07 cm x 27.94 cm)

POWER:
115/230 V ac, 50/60 Hz, 25 W nominal

3.3 OPERATION

3.3A INTRODUCTION

Operation of the Model 4144 is straightforward. Each of the operating parameters can be alphanumerically displayed at the front panel and quickly and easily set for optimum performance in the intended application. Once set, values remain in memory until changed, even if power is removed. Alternatively, the operating parameters can be set and the instrument's operation controlled from an external computer via the IEEE-488 Interface.

3.3B FRONT-PANEL CONTROLS

On/Off switch: The power switch is located at the upper left of the front panel. Note that, when the 4144 is powered down, it retains the settings in effect as of the moment of power-down. Those settings remain in effect until changed by the user.

Display: The 4144 features a 5 x 7 dot matrix alphanumeric display. Any adjustable parameter can be displayed and its value changed using the numeric keypad below the display.

Cursor: The Cursor switch to the right of the display selects the digit acted on by the keypad. The selected digit blinks steadily. Pressing the Cursor switch to the right causes the cursor (blinking position) to shift one place to the right. Pressing it to the left causes the cursor to shift one place to the left. If entering a multidigit number, the cursor will automatically shift to the right as each digit is keyed. However, it must be correctly positioned using the switch before you can begin keying the number. For example, if you wanted to specify a parameter value of "400", the cursor would have to be located at the third place to the left of the decimal point before you start keying the number.

Keypad: The keypad contains fifteen keys. Included are two Function keys, the digits 0 through 9, and the three operator keys, "+", "-", and "*".

The **Function** keys select the displayed parameter. The upper one moves the selection upwards through the stack and the lower one moves it downward. The stack folds so that pressing either key repeatedly eventually brings you back to the starting point.

The **Numeric** keys change the value of the displayed number at the cursor position. For example, if the number displayed at the cursor (blinking) position were "4" and you wanted a "9", it would only be necessary to press the "9" key. The new value goes into effect immediately; there is no separate **Enter** operation.

The "+" and "-" keys increment or decrement the number at the cursor position one step.

The effect of pressing the "*" key depends on the selected function, as follows.

1. If the **SAVE** function is selected (displayed), pressing the "*" key twice causes all of the parameters to be stored at the specified location.

2. If the **RECALL** function is selected, pressing the "*" key twice causes the parameter settings stored at the specified location to be recalled and immediately implemented.

3. If the **RUN SCAN** function is selected, pressing the "*" key once initiates **repetitively scanned** operation.

4. Pressing "*" initiates a **single scan** when the display indicates any of the following.

   SCAN INCREMENT
   TRIGGERS/STEP
   STEPS/SCAN

5. Pressing "*" initiates a **one-shot trigger** when the display indicates any of the following.

   CH A DELAY
   CH B DELAY
   RATE
   TRIGGER LEVEL

A B AB
T
GPIB ADDRESS

Trigger Mode: This multi-function switch allows the trigger-input circuits of the 4144 to be tailored to the available trigger source. You have the choice of four fixed and two adjustable levels. The fixed levels are:

LINE: Triggers internally at line frequency.

TTL: Triggers on positive-going edge of TTL trigger applied to Trigger Input connector.

ECL: Triggers on positive-going edge of ECL trigger applied to Trigger Input connector.

NIM: Triggers on negative-going edge of NIM trigger pulse (-16 mA into 50 Ω) applied to Trigger Input connector.

The two variable levels allow triggering on either the positive-going or negative-going edge of the trigger signal applied to the Trigger Input connector. The actual trigger level is set using the display and numeric keypad.

Triggered Light: Immediately to the right of the Trigger Selector switch is the **Triggered** light, which flashes briefly each time the 4144 is triggered internally or externally. At high repetition rates the light will appear to glow steadily when triggering is taking place.

Impedance Select: To the right of the Triggered Light is the **Trigger Input Impedance** toggle switch, which establishes the input impedance of the Trigger Input BNC connector at either 50 Ω or 1 MΩ.

Ext/Stop/Int Switch: In the **EXT** position, the 4144 triggers from an external source applied to the Trigger Input connector (or **line**) as selected by the Trigger Mode switch. In the **STOP** position, triggering is suppressed. In the **INT** position, the 4144 self triggers at the frequency set with the **RATE** function. Note that the External Trigger input remains "live" when **INT** is selected.

Trigger Input Connector: The external source is connected to this BNC connector. The input impedance is either 50 Ω or 1 MΩ, as selected.

Width Adjustments: A screwdriver adjustment adjacent to each of the three variable-width outputs, T_o, A, and B, sets the width of these outputs.

T_o Connector: The "Time Zero" pulse is provided at this connector. The leading edge of this pulse marks the start of the delay interval.

A Connector: The **A** Output pulse is provided at this connector.

B Connector: The **B** Output pulse is provided at this connector.

Pen Lift: A TTL Pen Lift output is provided at this connector. This output is at logic 0 during each scan. At the end of the scan it goes to logic 1 and remains there until the start of the next scan.

Ramp Output: During scanned operation a stepped ramp is provided at this connector. The ramp increments in amplitude each time the delay steps, reaching its maximum value of +10 V at maximum delay.

A.B: The signal provided at this connector is determined by the A and B outputs. The levels are 0 V and +5 V and you can select either sense. The first transition occurs on the leading edge of the A Output and the second on the leading edge of the B Output. The A.B function therefore provides a pulse with very low-jitter leading and trailing edges and width equal to the difference between the A and B delay settings. This output is completely independent of the A and B width settings. The start and finish occur a couple nanoseconds later than the A & B Outputs. The A.B width at half-height is typically within 2 ns of the A - B difference. The minimum width (half height) is nominally 7 ns and may of course be adjusted in position and width in 10 ps steps.

Use of the negative sense (A.B output down on A and up on B) may be restricted because an extra logic 0 pulse is present with this choice. The width of this pulse is about 0.5 μs and it is positioned just after the trailing edge of the output with the longest delay.

Note that the A.B output is not active in scanned operation.

Trigger Inhibit: A TTL logic 0 applied to this connector stops the 4144 from triggering the same as if the Trigger Mode switch were in the **STOP** position. This input is duplicated at the rear panel.

3.3C REAR PANEL

Other than for the IEEE-488 GPIB Interface and TRIG INHIBIT connectors, the only rear-panel connectora is the **SCAN INHIBIT** connector. This is an **active low** input that inhibits the outputs after the scan ends. If asserted, this input must go high for at least 100 ns to restart the scan. This input is active in scanned operation only.

3.3D DISPLAYED FUNCTIONS

CH A DELAY: In non-scanned operation, used to set the delay between the leading edge of the T_o output pulse and the leading edge of the **A** output pulse. In scanned operation, sets the initial delay. (Ch A only scans).

Note: A single internal trigger can be initiated by pressing "*" when this function is displayed.

CH B DELAY: Sets the delay between the leading edge of the T_o output pulse and the leading edge of the **B** output pulse. This is true for both scanned and non-scannned operation (Ch B does not scan).

Note: A single internal trigger can be initiated by pressing "*" when this function is displayed.

RATE: Used to set the Trigger Rate in **INT** triggered operation. **Note:** A single internal trigger can be initiated by pressing "*" when this function is displayed.

TRIGGER LEVEL: Used to set the input trigger level when operating in the External Trigger mode with either variable trigger level selected. The range is ±3 V and you can elect to trigger on either the positive or negative going slope.

Note: A single internal trigger can be initiated by pressing "*" when this function is displayed.

KEY ** TO SAVE AT ##: By setting the desired storage location (range is 1 to 30) and then pressing the "*" key twice, all of the parameter settings will be saved at the specified storage location.

KEY ** TO RECALL ##: By setting the desired storage location (range is 1 to 30) and then pressing the "*" key twice, the parameter settings stored at the specified location will be recalled and placed in effect immediately.

SCAN INCREMENT: This parameter specifies the value of the delay steps or increments in scanned operation. A single scan can be initiated by pressing the "*" key when this function is selected.

TRIGGERS/STEP: This parameter specifies the number of times the unit triggers at each delay value in scanned operation. After the specified number of triggers have occurred, the delay steps to the next value, and the cycle repeats. A single scan can be initiated by pressing the "*" key when this function is selected.

STEPS/SCAN: This parameter specifies the number of steps for a complete scan. The delay at the end of the scan is simply the **CH A DELAY** value plus the product of the **SCAN INCREMENT** value times the **STEPS/SCAN**. A single scan can be initiated by pressing the "*" key when this function is selected.

RUN SCAN: Selecting this function allows repetitive scanning to be initiated. Scanning is started by simply pressing the "*" key.

A B AB: Allows the output level for the **A OUTPUT** and for the **B OUTPUT** to be separately specified. The choices are +5 V, +10 V, and NIM (-16 mA into 50 Ω). The AB output levels are always 0 V and +5 V but you have the choice of **+** (+5 V pulse with 0 V baseline) or **-** (0 V pulse with +5 V baseline). The cursor is used to select **A**, **B**, or **AB** as appropriate and the "+" and "-" keys are used to make the parameter selections.

Note: A single internal trigger can be initiated by pressing "*" when this function is displayed.

T: Allows the level of the T_o output to be specified. The choices are the same as for the A and B outputs, +5 V, +10 V, and NIM (-16 mA into 50 Ω).

Note: A single internal trigger can be initiated by pressing "*" when this function is displayed.

GPIB ADDRESS: Allows the GPIB address to be set. This is the address that the controller must place on the bus prior to sending an instruction to the 4144.

Note: A single internal trigger can be initiated by pressing "*" when this function is displayed.

PEN LIFT DELAY IN/OUT: Allows delays to be inserted in the pen-lift function to facilitate analog plotting. The "+" key toggles this function.

When the Pen Lift Delay is OUT, the pen-lift output goes low coincident with the start of the scan and remains low for the duration of the scan.

When the Pen Lift Delay is IN, there is a one second delay between the pen-lift output going low and the start of the scan (beginning of the Ramp Output). At the end of the scan, the pen-lift goes high coincident with the end of the scan (ramp output plateaus). After a one second delay the Ramp Output begins to reset and three seconds are required to complete the reset.

3.3E TRIGGERING

Introduction: When the EXT-STOP-INT switch is in the STOP position, all outputs are inhibited. Similarly, a TTL logic 0 applied to the **TRIG INHIBIT** input also inhibits all outputs and prevents a scan from starting. In the EXT position, only external triggers or internal line triggers are accepted. In the INT position, external triggers are OR'ed with the output of the internal rate generator. As a result, unless you want to accept both trigger sources, any external trigger source should be disconnected when using the internal rate generator. Note that bias conditions for the internal trigger mode circuits are optimized with the input selector set to **ECL**. The frequency of the internal trigger generator is adjustable when the **RATE** function is selected.

External Triggering: In **EXT** triggered operation, triggering occurs on the positive-going edge of the applied waveform if **TTL** or **ECL** is selected, and on the negative-going edge of the applied waveform if **NIM** is selected. If either of the two variable positions is selected, the trigger level is determined by the **TRIGGER LEVEL** function (range nominally ±3 V). One of the

two variable trigger positions allows triggering on the positive-going edge of the applied waveform and the other allows it on the trailing edge. External triggers at rates up to 2 MHz will be accepted in normal operation as long as the interval between triggers is greater than the delay setting +350 ns for delays up to 80 μs.

Single Shot: The "*" key provides a single-shot trigger if the selected function is:

CH A DELAY
CH B DELAY
RATE
TRIGGER LEVEL
A B AB
T
GPIB ADDRESS

Triggering When Scanning: In scanned operation, triggers will be accepted and counted at rates as high as 20 kHz. If the maximum counting rate is exceeded during a scan, the counter will miss some triggers and will therefore deliver a larger number of output pulses per step. The accuracy of the output pulse delays will not, however, be affected.

3.3F FIXED DELAY OPERATION

In fixed delay operation from the front panel, each trigger (external, internal, or line) generates one delay cycle resulting in a T_0 pulse, an A pulse, and a B pulse. The A.B output is defined by the leading edges of the A and B outputs. The pen-lift and ramp outputs remain at 0 V during fixed delay operation. Fixed (as opposed to scanned) delay operation is the power-up default.

The fixed delay parameter values are set using the cursor switch in conjunction with the "+" and "-" keys for the applicable functions. These include:

1. **CH A DELAY:** Sets the Channel A output delay.

2. **CH B DELAY:** Sets the Channel B output delay.

3. **RATE:** Sets the trigger rate if triggering internally.

4. **TRIGGER LEVEL:** Applies only if selecting either of the two variable external trigger modes. If one of these is selected, this function sets the trigger threshold.

5. **A** +5 V **B** +5 V **AB+**
   +10 V +10 V **-**
   NIM NIM

   This function specifies the A, B, and AB outputs, as indicated. The baseline in all three cases is 0 V. For A and B, you have the choice of +5 V, +10 V or NIM (-16 mA into 50 Ω) outputs. The AB output levels are 0 V and +5 V and either sense may be selected.

6. **T** +5 V
   +10 V
   NIM

   This function specifies the T_0 output in the same way as the previously described function specifies the A and B outputs.

The **SCAN INCREMENT, TRIGGERS/STEP, STEPS/SCAN** and **RUN SCAN** functions have no relevance to fixed delay operation. The remaining functions, **KEY ** TO SAVE** and **KEY ** TO RECALL** don't set parameters but are active in both fixed and scanned operation for saving a set of parameters at one of the thirty storage areas or for recalling a set of parameters from a storage area. Similarly, the **GPIB ADDRESS IS ##** function is active in both fixed and scanned operation but has no significance unless the 4144 is to be controlled from an external computer.

3.3G SCANNED OPERATION

In scanned operation, the delay isn't fixed but rather is stepped, starting at an initial value and incrementing in equally spaced steps to sweep the delay over an interval defined by user specified parameters.

With the exception of the **CH B DELAY** function (Channel B does not scan), the parameters for fixed delay operation apply also to scanned delay operation. The **CH A DELAY** function differs a bit in that, instead of setting a fixed delay, it sets the initial value of the scan delay range. Other functions unique to scanned operation are as follows.

1. **SCAN INCREMENT:** Specifies the delay step size, that is, the amount by which the delay will increase each time it increments. The delay does not increase linearly but rather in steps, with a specified number of triggers at each step. A single scan can be initiated by pressing the "*" key when this function is selected.

   Setting this parameter to "0" allows burst operation in which a burst of pulses at the trigger rate is generated. The number of pulses in each burst is equal to the TRIGGERS/STEP and a new burst occurs every 1 ms.

2. **TRIGGERS/STEP:** Each trigger initiates an output cycle and each is counted. When the number counted equals the TRIGGERS/STEP parameter, the delay steps to its next value. A single scan can be initiated by pressing the "*" key when this function is selected.

3. **STEPS/SCAN:** This parameter determines the number of steps in a scan and hence a scan's maximum delay. A single scan can be initiated by pressing the "*" key

when this function is selected.

4. **RUN SCAN:** Repetitively scanned operation can only be initiated (press "*" key) when **RUN SCAN** is selected.

Burst Mode: Burst operation is a subset of scanned operation and is established by simply setting the **SCAN INCREMENT** parameter to "0". When the "scan" is started, bursts of output pulses at the trigger rate will be generated. The number of pulses per burst will equal the **TRIGGERS/STEP** parameter and successive bursts will be 1 ms apart. If scanning repetitively, bursts will continue to be produced for as long as scanned operation is maintained. In single scan operation the number of bursts equals the Steps/Scan setting.

Burst operation is particularly well suited to an experiment that calls for a specific number of output pulses. For example, if you needed a burst of ten pulses, you would set the **SCAN INCREMENT** parameter to "0", the **STEPS/SCAN** parameter to "1", and the **TRIGGERS/STEP** parameter to "10". Then select any scan function except **RUN SCAN** to program single-scan operation. When the "*" key is subsequently pressed, a single burst of pulses will be generated at the trigger rate.

Single vs Repetitive Scanning: The choice of displayed function determines whether single scanned or repetitively scanned operation will take place when the "*" key is pressed. If **RUN SCAN** is the selected function, scanning will be repetitive, that is, one scan will follow another automatically until halted by pressing any key except "*". If any of the other scan functions (**SCAN INCREMENT, TRIGGERS/STEP,** or **STEPS/SCAN** is selected, a single scan will be generated each time "*" is pressed.

Typical Operation: An example of typical scan mode operation follows. Suppose you needed to scan repetitively from 5 μs to 25 μswith a 1 kHz trigger rate. Further assume that you want the delay to be stepped 50 times per scan in increments of 400 ns and with 20 triggers at each delay value.

You would probably begin with fixed delay operation (internal trigger mode and rate set to 1000 Hz) to set the output pulse width. The next step would be to set the Scan Functions as follows.

CH A DELAY	**5000 ns**
SCAN INCREMENT	**400 ns**
TRIGGERS/STEP	**20**
STEPS/SCAN	**50**

Repetitive scanning as programmed could then be initiated by selecting the **RUN SCAN** function and pressing the "*" key. If you were to monitor the A output pulse with an oscilloscope triggered from the T_0 output, you would see the pulse move smoothly and repetitively across the oscilloscope screen from a point 5 μs after the trigger to a point 25 μs after the trigger. Scanning would continue until terminated by pressing any key except "*". When a scan is terminated or completed (single-scan operation), no further triggering occurs unless a new scan is initiated or fixed delay operation is restored.

To restore fixed delay operation, step through the functions to **CH A DELAY** or **CH B DELAY**. Normal fixed delay operation will commence immediately.

3.3H GPIB INTERFACE

Introduction: A rear-panel IEEE-488 GPIB interface allows control from an external computer. Ordinarily this would be a personal computer equipped with a GPIB interface card such as the General Instruments PC2A. The interface card satisfies the GPIB protocol and connector requirements. The user's concern is to write the necessary control software that will send the correct GPIB address (set at the front panel) and the necessary commands.

The 4144 supports the LISTENER state (only), allowing all critical parameters and functions to be set from a remote location. The syntax depends on whether the command takes an operand. If it doesn't, simply send the command character followed by a line feed. If the command does take an operand, send the command character, followed by a line feed, followed by the command character **again**, and then the operand and another line feed. The number of digits in the operand must be as specified in Table 3-1. Use leading zeros to pad out the operand to the required number of digits if necessary.

Table 3-1 lists the 4144 commands.

Code Sequences: The following ASCII code sequence would set the CH A delay setting.

A <LF> A d1 d2...d10 <LF>

The initial function designator is the ASCII character **A** followed by a line feed. The **A** designates **Function A** (Channel A Delay). The second **A** respecifies Function A and the digits that follow specify the actual delay value and are accepted right justified. As indicated in the table, the last two digits sent represent the two places to the right of the decimal point. A final line feed ends the command. No decimal point is used. For example, a delay of 63.26 ns in Channel A would be specified by the sequence:

A <LF> A 0 0 0 0 0 0 6 3 2 6 <LF>

Notice that leading zeros are used to obtain the required

COMMAND	MAX # DIGITS	NUMBER OF DIGITS TO RIGHT OF DECIMAL POINT	COMMAND
CH A DELAY	10	2	A
CH B DELAY	10	2	B
RATE	10	3	E
PEN LIFT DELAY OUT	1	none	F0
PEN LIFT DELAY IN	1	none	F1
TOGGLE PEN DELAY	none	none	F
SCAN INCREMENT	10	2	G
TRIGGERS/STEP	5	none	H
STEPS/SCAN	4	none	I
START REPET. SCANS	none	none	J
START SINGLE SCAN	none	none	K
ABORT SCAN	none	none	any character

Table 3-1. MODEL 4144 COMMAND SET

ten places.

If the same command is repeated, the initial function designator may be omitted after the first command. For example, to repeatedly control the Channel A delay you could send:

A <LF> A d1 d2 d3...dn <LF>
A d1 d2 d3...dn <LF>
A d1 d2 d3...dn <LF> etc.
As shown, the first time the command is sent, the command character is sent twice. The second and subsequent times, it is only sent once and is directly followed by the operand. This, of course, only applies if the command is repeated directly. If another command is interposed between repetitions, then the full syntax requirements must be observed.

Saving Parameter Settings: The current settings are saved each time a front-panel key is pressed or the cursor switch is activated. These settings are retained indefinitely when power is removed.

This process is, by definition, automatic when settings are entered with the front-panel keys. To store remotely loaded parameters as the current set so that they will be restored at the next power on, simply activate the front-panel cursor switch prior to powering down. Remotely loaded parameters may also be stored in the same way as manually entered parameters in any of 30 locations by using the front-panel **SAVE SETTINGS** function.

Note that there is a pause before the scan starts when the **J** or **K** command is applied after making a parameter change. The length of the pause will be approximately 10 ms. If no manual or remote command change has been made since the last time **K<LF>** was applied, the scan will begin in less than 5 ms after applying the **K<LF>** command sequence.

GPIB Serial Poll: The Model 4144 responds to a GPIB serial poll. Bits 0 (least significant bit) and 1 have meaning as follows.

Bit 0: SCAN ACTIVE - This bit is set while the scan is active. It is useful for synchonizing the host computer with the 4144. The computer can cause Trigger Inhibit to be held LOW, send the START SCAN command, and poll the 4144 while waiting for this bit to be set. As soon as Bit 0 set is detected, the computer can allow Trigger Inhibit to be released.

Bit 1: PARAMETER ERROR - Set when a parameter sent via the GPIB is invalid.

Chapter 4
Model 4161A

4.1 INTRODUCTION

The Model EG&G PARC 4161A 2-Channel Analog-to-Digital Converter gives the 4100 Boxcar System display and conversion capabilities. With two independent channels, each with its own signal and trigger input, the 4161A can read and convert the output of two Model 4121B's. The front-panel 3-1/2 digit display allows the level in either channel to be conveniently read. With its analog input and meter, the 4161A can also easily read analog voltages too.

Both the 4161A and the 4144 Digital Delay Generator feature computer control via an IEEE-488 GPIB interface for complete system control using the Boxcar Applications Software. The 4161A additionally has an RS232C Interface, giving it an alternative means of control.

4.2 SPECIFICATIONS

Module Type: Dual Width NIM Bin

ADC Specs:

Number of Channels: 2

Number of Bits: 11 bits plus sign

Range: ±10 V

Resolution: 5 mV

Accuracy: ±5 mV

Linearity: ±5 mV

GPIB Indicator LED's: TALK, LISTEN, SRQ, & REMOTE

Digital Display: 3.5 digit LED, switch selectable to either channel.

Analog Meter: Center Zero; Edge mount; 5% of full scale accurate.

ADC Signal Inputs: Separate BNC connector for each channel.

Sensitivity: ±10 V f.s.
Input Impedance: 1 MΩ

ADC Trigger Inputs: Separate BNC connector for each channel; TTL. Triggers on leading edge of applied positive logic TTL pulse.

Reset: Pushbutton switch accessible through small hole in front panel.

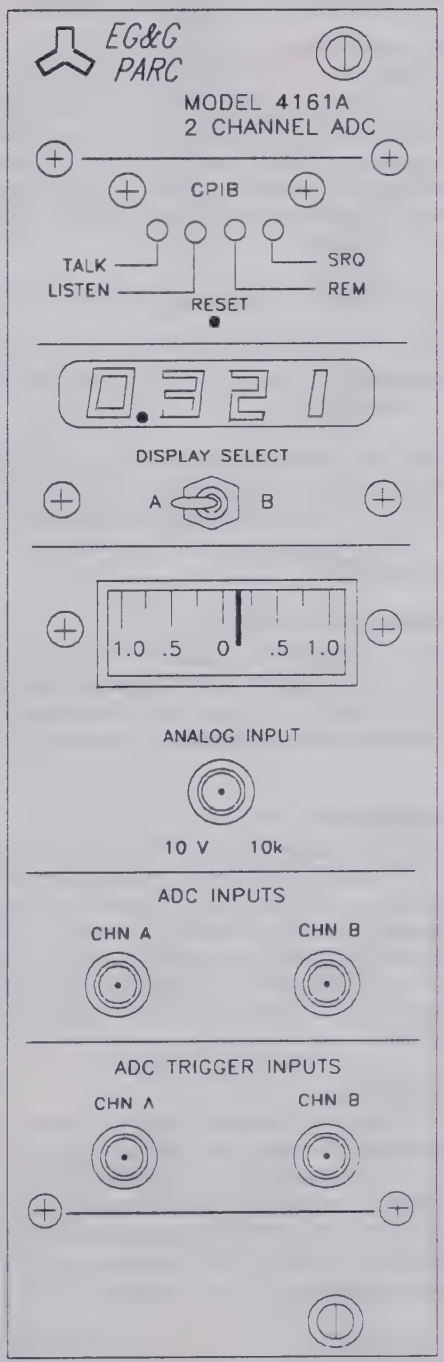


Figure 4-1. MODEL 4161A

Busy: Rear-panel BNC connector open-collector TTL output. **Busy** of 4161A and 4121B's are tied together and applied to **Trigger Inhibit** input of 4144 so that system triggering is inhibited when any module is still active in response to previous trigger applied to 4144.

Control Interfaces: IEEE-488 GPIB and RS232C.

Data Rate: The maximum data rate can vary over a wide range according to the computer hardware and depending on whether synchronous or non-synchronous operation has been selected. Operating on the IEEE-488 GPIB, the maximum ASCII rate is typically in the range of 100 readings per second to about 150 readings per second. Much higher data rates have been obtained using binary dump techniques.

Data Format: integer format; ±2048 mV f.s.; ±5 mV resolution.

Bin Requirement: The Model 4161A receives its power from a 4152 Power Bin.

> **Note:** The bin without a power supply is the Model 4152/99. With the power supply installed it is simply a Model 4152. The Power Supply is the Model 7018.

4.3 INSTALLATION

4.3A INCOMING INSPECTION

Newly received apparatus should be inspected for shipping damage. If any is noted, immediately notify EG&G PARC and file a claim with the carrier. The shipping container should be saved for inspection by the carrier.

4.3B BIN REQUIREMENTS

The Model 4161A Display - ADC Module is a standard double-width Nuclear Instrument Module (NIM). As such, it requires a standard NIM bin that conforms to the NIM standard as defined in AEC report *TID-20893 (Rev), Standard Nuclear Instrument Modules*. EG&G PARC provides the Model 4152, a six-wide bin equipped with the Model 7018 Power Supply, as a component of the 4100 system.

4.3C GROUNDING

The 4161A's chassis is internally returned to the third wire or grounding pin of the standard bin's input power connector. For equipment and operator safety, it is important that this grounding pin be returned to a good earth-ground connection via the ground wire of the power distribution system. This precaution serves to prevent shock hazards.

4.3D VENTILATION

A standard NIM bin does not incorporate forced air ventilation.

4.4 FRONT PANEL

4.4A GPIB INDICATORS

TALK: Indicates that the 4161A has data ready to send to the host computer. It doesn't extinguish until it has finished sending this output. It is **not** a signal that the host computer has transmitted the GPIB **TALK** message.

LISTEN: Lights when the first character of command has been sensed. Does not extinguish until the terminator is sensed. This is **not** a signal that the host computer has transmitted the GPIB **LISTEN** message.

SRQ: Lights when the Model 4161A makes a service request (SRQLO asserted). It remains lighted until the computer competes a serial poll of the Model 4161A.

REM: Indicates that RENLO has been asserted and the listen address applied. It does not indicate that the front panel is locked out, nor does it indicate a change in the communications receptiveness of the system. The front panel and both interfaces are always "alive".

4.4B DIGITAL DISPLAY

This 3-1/2 digit display can show the output of either channel as selected by the associated Display Select toggle switch. The left-most digit is either unlighted or displays a "**1**". The other three digits can takes values of **0** through **9**. Full scale display is **1.023** or **-1.024**, according to the polarity, and represents **10.23 V** or **-10.24 V** respectively. The decimal point is fixed.

4.4C DISPLAY SELECT SWITCH

This toggle switch, located directly beneath the Digital Display, selects which of the two channels, **A** or **B**, is displayed.

4.4D ANALOG METER

This meter is completely independent of the A/D converter functions. It monitors the voltage applied to the associated **Analog Input** BNC connector, which has an input impedance of 10 kΩ. The full-scale sensitivity is ±10 V.

4.4E ADC INPUTS

There are two, one for **Channel A** and the other for **Channel B**. The analog signals to be converted should be applied to these connectors. When the 4161A is operated as a component of the 4100 System, the **LSO** connector of the "**A**" 4121B would be cabled to the 4161A **Chn A ADC Input** and the **LSO** connector of the "**B**" 4121B would be cabled to the 4161A **Chn B ADC Input**.

4.4F ADC TRIGGER INPUTS

There are two, one for **Channel A** and the other for **Channel B**. Each time a TTL trigger is applied, an A/D conversion of the voltage applied to the corresponding Signal Input is initiated. The 4161A conversion time is approximately 100 μs. However, in a real computer-controlled experiment, this figure has little relevance, the

pacing factor being the maximum speed at which the computer can take readings over the IEEE-488 GPIB, typically about 150 readings per second. **Caution:** Because of the trigger overhead requirements, as the trigger rate is increased, less time remains to handle the GPIB data transfer. The result is a progressive decrease in the **data transfer rate** as the trigger rate is increased beyond a certain point. **Do not attempt to maximize the data transfer rate by applying triggers at a much higher rate than the 4161A can handle.**

4.4G RESET

There is a small pushbutton switch accessible through a small hole in the front panel just below the GPIB indicators. Pressing this pushbutton resets the 4161A logic, including the A/D converters. A paper-clip makes a suitable tool for actuating this switch. There would be no need to operate this switch in a typical operating session.

4.5 REAR PANEL (Figure 4-2)

4.5A RS232C CONNECTOR

This interface allows the Model 4161A to be controlled from an RS232C source. Note, however, that the 4144 Digital Delay Generator has an IEEE-488 GPIB interface only. As a result, complete system control, as opposed to control of the 4161A only, is only possible via the IEEE-488 GPIB bus.

4.5B GPIB CONNECTOR

Computer control via the IEEE-488 GPIB is done using this connector. It requires that a GPIB cable be connected from the computer's IEEE-488 GPIB port to this connector. In a system that incorporates the 4144 Digital Delay Generator, you would additionally need another cable to connect the 4144 to the bus.

4.5C BUSY OUT

This open-collector TTL output goes to logic 0 whenever the 4161A is active. In a 4100 System, this output is tied to the **BUSY OUT** connectors of each 4121B and to the **Trigger Inhibit** input of the 4144. As a result, when the 4161A or either 4121B is busy, the 4144 is held off and cannot be triggered.

4.5D NIM CONNECTOR

The power and ground connections are made via this connector, as follows.

PIN	USE
11	-6 V
17	-12 V
28	+24 V
29	-24 V
34	GROUND
16	+12 V
10	+6 V

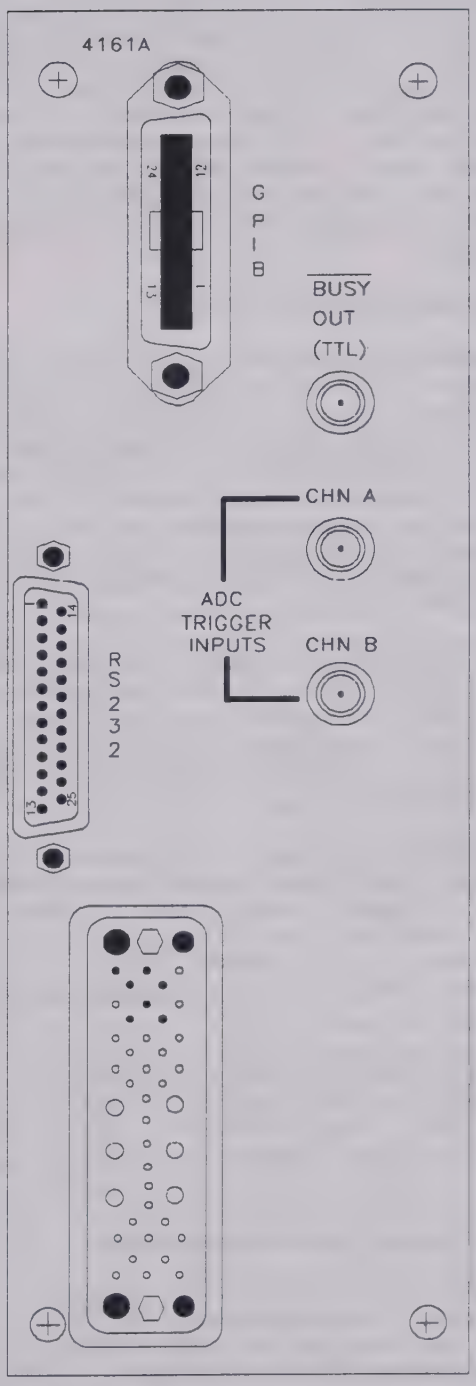


Figure 4-2. BACK PANEL

4.6 OPERATION

4.6A INTRODUCTION

Operation of the Model 4161A is extremely simple. Each time a trigger is applied to either of the two channels, the voltage at the corresponding ADC Signal Input

connector undergoes an analog-to-digital conversion. The resulting number is latched so that it can be conveniently read via a host computer. Computer controlled operation is particularly straightforward using the EG&G PARC Boxcar Applications Software. With respect to the front panel, other than to connect the cables as previously described, there are no operator considerations other than the option of displaying the value in either channel at the front panel. The front-panel Display Select switch allows you to choose **A** or **B**, whichever you wish to have displayed.

4.6B COMPUTER CONTROL

Terminator: The Model 4161A accepts either <CR> or <CRLF> as a command terminator. In turn, the terminator it appends to a transmitted data value, either <CR> or <CRLF>, is set by Switch 8 of internal DIP Switch Assembly S1 (Figure 4-3). The selected terminator must be acceptable to the host computer. It is not always easy to check reference documentation to determine a computer's terminator requirement. Where this is the case, one can simply try first one possibility and then the other. One will work.

GPIB DIP Switch Assembly S1: This switch assembly sets the GPIB address. It also allows selection of the GPIB Test Echo function, as well as selection of the terminator to be appended to transmissions from the 4161A to the computer in both GPIB and RS232C communications.

The GPIB differentiates between instruments on the bus by means of the address the controller places on the bus before sending an instruction. Only the instrument having the corresponding address will respond to the instruction. The address is set by switches 1 through 5 of the S1 Assembly. The value weighting for each is shown in Figure 4-3.

When the Test Echo function is ON, every character transmitted or received via the GPIB port is echoed to the RS232C Interface connector. This function is particularly useful in program development.

Switch 8 of the S1 Assembly selects the terminator, which functions as previously described.

Commands: The Model 4161A recognizes twelve commands. A valid transmission to the 1461 consists of one of these commands followed by a terminator, either Carriage Return **<CR>** or Carriage Return Followed by a Line Feed **<CRLF>**. These commands can be applied via either the GPIB or RS232C Interface, both of which are always active. The function of each command is as follows.

RDA: Reads the output of Channel A and sends the equivalent ASCII string, followed by a terminator, to the host computer. With a ±f.s. input applied, the transmitted ASCII string is 2047 or -2048. The number returned is the number of A/D counts, and is double the number displayed on at the front panel. Each count is 5 mV. An ASCII response consists of the numeric digits, followed by a space character, followed by the terminating character.

Example: <digit><digit><digit><space><CR>
(i.e., "255 <CR>")

Note that **RDA** is a **non-synchronous** read command, that is, it operates independently of the 4144 Digital Delay.

Binary transfers are also possible. See discussion of **BIN1** and **BIN0** commands.

RDB: Like the RDA command except that it applies to Channel B.

RAB: Reads the output of both Channel A and Channel B and sends the readings to the computer. The format of the response is:

<digit><digit><space>,<digit><digit><space><CR>
(e.g., "34 ,82 <CR>")

Like RDA and RDB, this command is **non-synchronous**.

Synchronized Operation: In scanned operation with a 4144, the previously described commands are not suitable for acquiring and transferring data rapidly. The problem is that the 4144 Delay Generator can be triggered before transfer of the result of the previous A/D conversion is complete, causing synchronization to be loss. The commands described in the following paragraphs remedy this problem by using the BUSY line to control the trigger receptiveness of the 4144 in such a way as to keep the 4144 and transfer of the data to the host computer synchronized. There will be one A/D Conversion and one data transfer to the computer for each trigger accepted by the 4144.

BSYON: This command (BUSY ON) causes the BUSY output to go low, inhibiting the 4144's trigger input. It remains low unless:

1. **BSYOFF** is applied, restoring non-synchronized BUSY control, or:

2. One of the synchronized-mode read commands, **SA**, **SB**, or **SAB** is applied. Each time this happens, BUSY goes high, allowing the 4144 to receive a trigger. The sample is taken and the 4161A is triggered. BUSY immediately goes low so that the 4144 can't be triggered again, allowing the 4161A as much time as it needs to complete the conversion and for the resulting value to be transferred to the computer.

   Note: In non-synchronized operation, the 4161A holds

BUSY low only from the arrival of the A/D Trigger to completion of the A/D conversion. BUSY is not held down during data transfer to the computer.

BSYOFF: Restores normal (non-synchronized) BUSY control.

SSF: Initializes the 4161A for synchronous-mode operation. The SSF command must be applied after the **BSYON** command to establish synchronous operation. Once it has been applied, any number of SA, SB, or SAB commands may be used to read the data during the experiment.

CSF: This command tells the 4161A to exit the synchronous mode. It is normally applied after the last SA, SB, or SAB command and before the BSYOFF command.

SA, SB, & SAB: These commands are identical in every way to the RDA, RDB, and RAB commands, except that they additionally control the BUSY output in synchronous operation. BUSY is held low (BSYON) until one of these three commands is received. When the command is recognized, BUSY is released until an A/D trigger input is received, at which time BUSY is pulled low again. BUSY remains low until the next synchronous read command is received. Compare this to non-synchronous operation in which the 4161A holds BUSY low only from the time the A/D Trigger is detected to completion of the subsequent A/D conversion.

Note that proper synchronization requires that the germane commands be used together and in a specific order, as follows.

BSYON: Forces BUSY low.

Send a scan-start command to 4144. The scan will be held off because BSYON has been applied.

SSF: Establishes synchronized mode.

begin loop

SA, SB or **SAB:** Read and store A/D conversion value.

end loop

CSF: Ends synchronized mode.

BSYOFF: Re-establishes normal BUSY control.

X: Initiates the binary data transfer mode for Channel A. This mode allows the system to take data and transfer it to the computer as rapidly as possible. Applying the command **once** allows an indeterminate number of readings to be taken. Readings continue to be taken until the mode is terminated, accomplished by sending any character to the 4161A. The data rate

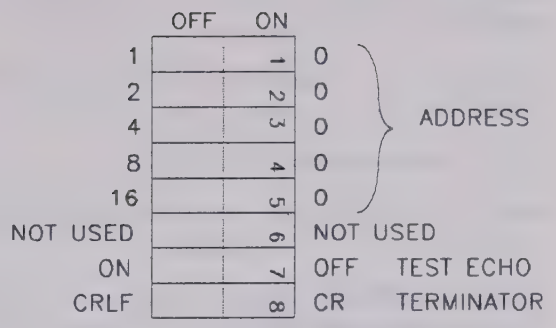


Figure 4-3. GPIB SWITCH ASSY

achieved depends on several factors, including the kind of computer hardware used, and can vary over a wide range.

Each reading is transferred as two eight-bit binary bytes, with the high-order byte transferred first. The data, eleven bits plus sign, occupies all of the high-order byte and the four highest bits of the low-order byte. The host computer has to do an unsigned shift of the data four places to the right to remove the four low bits of the low-order byte. With a +f.s. input applied, the transferred number (after removing the four low bits of the low-order byte) is **111111111111**. With a -.f.s. input applied, the transferred number (after removing the four low bits of the low-order byte) is **000000000000**. The decimal equivalents of these two values are **+4096** and **0**. Thus, to convert to a bipolar representation, the computer must subtract **2048** from the decimal equivalent of the transferred value. **Note:** No terminators are sent in the fast-transfer mode.

During data transfer, **BUSYLO** holds off the 4144. As soon as the transfer is complete, the 4144 becomes trigger sensitive again. If a trigger is immediately applied, or if a continuous stream of triggers at a higher rate is applied, the highest read/transfer rate can be achieved.

Y: Like the **X** command, except that the non-synchronous fast transfer mode is established for Channel B.

BIN1: This command activates the binary transfer mode, in which the response format for the RDA, RDB, SA, & SB commands is:

<low_byte> <high_byte>

The GPIB EOILO line is asserted on the second byte. Note that no shifting or other adjustments are needed; full-scale is already ± 2048. Somewhat higher data transfer rates can be achieved in the Binary Transfer Mode than in the ASCII Transfer Mode. **Caution:** To get the fastest possible transfer rate, the front-panel display and LED's are turned off in the Binary Transfer Mode, which might cause the user to inadvertently infer that the 4161A has lost power.

Similarly, the response format for the RAB and SAB commands becomes:

<low_byte_A> <high_byte_A> <low_byte_B> <high_byte_B>

EOI$_{LO}$ is asserted on the fourth byte.)

BIN0: Transfers the 4161A from the Binary Transfer Mode to the ASCII Transfer Mode.

DSPON and **DSPOFF:** These two command control the front-panel display and LED's. DSPOFF turns the display off. DSPON turns it on. Note that DSPOFF is automatically established in the Binary Transfer Mode.

4.6C GPIB COMMAND OVERRUNS AND SERIAL POLLING

For successful GPIB control, it is essential that you not send a command to the 4161A while the previous command is still being executed. One can easily determine whether the 4161A is ready for a new command by performing a serial poll and then reading bits 0 (COMMAND DONE) and 7 (OUTPUT READY) of the Serial Poll Status Byte. A serial poll can be performed at any time. The bit weighting of the Serial Poll Byte is shown in Table 4-1.

Since the COMMAND DONE and OUTPUT READY bits are provided to facilitate orderly communications between the instrument and the host computer, it may be instructive to consider their behavior during a typical command operation.

1. Quiescently, COMMAND DONE is asserted and OUTPUT READY is cleared. The instrument is not busy executing a previously applied command and it has no output ready to transmit to the computer.

2. COMMAND DONE is cleared when the instrument detects the first character of an applied command.

3. Assuming the command was one requiring transmission of data to the computer, some unspecified time later OUTPUT READY will be asserted.

4. OUTPUT READY remains asserted (and, consequently, COMMAND DONE remains cleared) until the data is actually sent. Specifically, just before the very last character is sent, OUTPUT READY clears. Shortly afterwards, COMMAND DONE gets asserted again, restoring the initial conditions.

4.6D RS232C INTERFACE

As previously stated, 4161A commands can also be applied via the RS232C port. To be assured of successful RS232C communications, you have to satisfy RS232C handshaking and parameter matching requirements as explained in the following paragraphs.

RS232C PARAMETER SELECTION

The RS232C Standard specifies many parameters, such as voltage (logic) levels, transfer rates, etc. Some parameters have optional values that are user selectable. In the case of the Model 4161A, these selections are made via internal DIP switch assembly S2, as shown in Fig. 4-4.

BIT	MEANING	COMMENT
BIT 0 (LSB)	COMMAND DONE	Bit 0 is cleared when the Model 4161A is busy executing a command. It is set when execution of the previous command is complete.
BIT 1	COMMAND ERROR	Bit 1 is set when a command error occurs. It is cleared when the next valid command is issued.
BIT 2	NOT USED	
BIT 3	NOT USED	
BIT 4	A/D OF A DONE	RDA command can be applied.
BIT 5	A/D OF B DONE	RDB command can be applied.
BIT 6	SRQ	Bit 6 is set when service is being requested. It is cleared when service is not being requested.
BIT 7	OUTPUT READY	Bit 7 is set when the Model 4161A has output ready to dump to the computer. It is cleared when the 4161A has no output to dump.

Table 4-1. SERIAL POLL STATUS BYTE BIT WEIGHTING

The user has the choice of either seven or eight data bits; even, odd, or no parity (parity is always ON with 7 data bits and always OFF with 8 data bits); and either one or two stop bits. These parameters must be selected to be compatible with the host computer. Note that the combination of eight data bits and two stop bits is recommended for operation at speeds greater than 9600 baud. The use of two stop bits increases communication reliability by making it less likely that the receiver and transmitter will get out of synchronization. When transmitting eight data bits, the eighth (most significant) bit is transmitted first. This bit must be received as a '0' for ASCII characters.

Note that the baud rate is set by switches 5, 6, & 7 of the S2 DIP switch assembly, as follows.

Baud Rate	#7 (msb)	#6	#5 (lsb)
110	OFF	OFF	OFF
300	OFF	OFF	ON
600	OFF	ON	OFF
1200	OFF	ON	ON
2400	ON	OFF	OFF
4800	ON	OFF	ON
9600	ON	ON	OFF
19600	ON	ON	ON

An Echo ON/OFF selection is also provided. If Echo ON is selected, each character received via the RS232C port will be echoed back to the character source. The RS232C echo is normally used only when the 4161A is connected to a CRT terminal.

Other factors to consider include details of the pinout and the choice of terminator, and the prevention of command overruns. The terminator considerations are the same as for GPIB communications, that is, the terminator must be selected to match the requirements of the computer, and the selection is made via the internal DIP switches as previously described.

Users should be careful not to confuse the two echo functions. The RS232C Local Echo (FUNCTION F5) allows each character received via the RS232C port to be reflected back to its source. On the other hand, the Test Echo (FUNCTION F9) reflects to RS232C port 1 every character that passes in or out of the GPIB port. This feature is primarily used during program development or in system troubleshooting.

HANDSHAKING

The user has the choice of CLEAR-TO-SEND or XON/XOFF control of RS232C data transmission between the 4161A and the host computer

If CLEAR-TO-SEND control is selected, CLEAR-TO-SEND must be asserted before the 4161A will transmit. (If CTS is left unconnected, it will float to the "asserted" state.) The data stream can be cut off by making

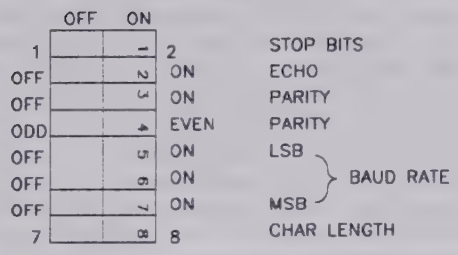


Figure 4-4. RS232C PARAMETERS

CLEAR-TO-SEND negative. Making CLEAR-TO-SEND positive restores the data flow. If CLEAR-TO-SEND is made negative, the Model 4161A will finish transmitting the character being processed at that moment, and may transmit one or two additional characters as well. The computer's software must anticipate the additional characters.

Note that CLEAR-TO-SEND is returned to +12 V through a pull-up resistor inside the Model 4161A so that it is allowed unimpeded transmission if the line is left floating.

When setting up CLEAR-TO-SEND control, the pin connections will depend on whether the computer's RS232C interface is configured as a DTE or DCE device. REQUEST-TO-SEND and CLEAR-TO-SEND are complementary handshake lines. In a DTE device, such as the Model 4161A, CLEAR-TO-SEND is an input and REQUEST-TO-SEND an output. Thus, if the computer's interface is also a DTE device, pin 4 of the 4161A would connect to pin 5 of the computer, and pin 5 of the 4161A would connect to pin 4 of the computer. On the other hand, if the computer's interface is configured as a DCE, even though the names of these lines are the same, their function is reversed, that is, CLEAR-TO-SEND is an output and REQUEST-TO-SEND is an input. With respect to connecting the 4161A to a computer that acts like a DCE device, pin 4 of the 4161A would connect to pin 4 of the computer and pin 5 of the 4161A would connect to pin 5 of the computer.

If XON/XOFF control is selected, the "characters" XOFF (CONTROL S; ASCII decimal 19) and XON (CONTROL Q; ASCII decimal 17) are used to control RS232C data transmission. XOFF will prevent RS232C (only) transmissions. If applied during a transmission, the transmission will halt. However, the host computer must be ready to accept a couple more characters after it transmits XOFF. XOFF does not prevent characters from being echoed if the ECHO mode is ON. Transmission resumes when XON (CONTROL Q) is applied.

TRANSMIT/RECEIVE PIN ASSIGNMENTS

With respect to the RS232C standard, the Model 4161A behaves like a Data Terminal Equipment (DTE) device, and as such, transmits data on pin 2 and receives data on pin 3. The RS232C port on most computers is configured the same way. When this is the case, pin 2 of the 4161A's RS232C connector should connect to pin 3 of the computer's RS232C connector, and pin 3 of the 4161A's connector should connect to pin 2 of the computer's RS232C connector. (This is often referred to as a "modem-eliminator" configuration.) However, if the computer is configured as a Data Communications Equipment (DCE) device, pin 2 at the 4161A should connect to pin 2 at the computer, and pin 3 at the 4161A should connect to pin 3 at the computer.

AVOIDING COMMAND OVERRUNS

Command overrun errors (caused by sending a new command while the 4161A is still busy executing the previous command) are prevented by using the serial prompt function provided. Simply put, each time a command is sent to the Model 4161A, the computer should wait until it receives the prompt (asterisk or question mark) before it sends the next command. (The 4161A sends the prompt when it is ready for a new command.) If the previously executed command generated no errors in the 4161A, the prompt character will be an asterisk (*). If the previously executed command did generate an error in the 4161A (such as 'Invalid Command', etc.), the prompt is a question mark (?). The prompt is transmitted after power up and after completion of every command. By waiting for the prompt character, Command Overrun Errors can be avoided.

LOGIC LEVELS

RS232C TRANSMIT AND RECEIVE LEVELS: The voltage levels defined in the RS232C Standard are:

DATA:	Logic 0 is +3 V to +25 V. Logic 1 is -3 V to -25 V.
CONTROL:	Asserted is +3 V to +25 V. Unasserted is -3 V to -25 V.

One might infer from these standard levels that the Model 4161A's RS232C output levels will be on the order of +25 V. This is not the case. The actual output levels are +12 V, and the input levels MUST be less than +15 V (higher levels will damage the 4161A).

OTHER REQUIREMENTS

Other factors to consider when establishing serial communication include details of the port pinout (provided in Appendix C) and the choice of terminator character. The terminator considerations are the same as those used for GPIB communications, that is, the terminator selected, either CARRIAGE RETURN <CR> or CARRIAGE RETURN followed by a LINE FEED <LF> must be selected to match the requirements of the host computer.

TROUBLESHOOTING RS232C COMMUNICATIONS PROBLEMS

1. Check the wiring of the RS232C connector. The Model 4161A transmits on pin 2 and receives on pin 3. From the point of view of the RS232C standard, it behaves like a DTE device. RS232C interface lines are always named from the perspective of a DTE. In the case of a DCE, this can cause confusion because the function of the line may be reversed. For example, in both a DTE and a DCE, pin 2 is called the "TRANSMIT DATA" line and pin 3 the "RECEIVE DATA" line. However, whereas a DTE transmits on pin 2 and receives on pin 3, a DCE receives on pin 2 and transmits on pin 3. This is the reason why the connections to pin 2 and 3 must sometimes be reversed before a RS232C link will work correctly.

   There is a similar relationship between REQUEST-TO-SEND (pin 4 in both DTE and DCE devices) and CLEAR-TO-SEND (pin 5 in both DTE and DCE devices). In the case of a DTE (Model 4161A), CLEAR-TO-SEND is an input and REQUEST-TO-SEND is an output. If the RS232C interface of the computer is configured as a DTE, pin 4 of the 4161A connects to pin 5 of the computer, and pin 5 of the 4161A connects to pin 4 of the computer. If the computer's RS232C interface is configured as a DCE, the functions are reversed. In this case, pin 4 of the 4161A connects to pin 4 of the computer and pin 5 of the 4161A connects to pin 5 of the computer. The 4161A's serial port pinout is provided in Appendix D.

2. Check that the following settings agree on both the 4161A and host computer.

   BAUD rate

   Number of Character Bits in the Serial Data Word.

   Logic state of the eighth character bit, if selected.

   Parity Settings.

   Number of Stop Bits.

3. Check that the host computer and 4161A agree on the terminator character, which should be either <CR> or <CRLF>.

4. Avoid command overruns by making the host wait for the RS232C prompt character before sending a new command.

5. Check that there are no timing problems. Two different timing problems are commonly encountered when RS232C is used to interface a host computer with a Model 4161A. Both are associated with responses from the 4161A to the host.

First, when the host sends a command to the 4161A, the host must be ready to accept the instrument's response before the 4161A sends the first character of that response. There are three methods for solving this problem. The host's program can be written to have a fast turnaround from the portion of it that sends the command to the portion that receives the response. This is not always possible with a slow, high-level language. It may work at low baud rates, but it is not a guaranteed solution.

Alternatively, the host can use its Request-To-Send line to inhibit the 4161A's Clear-To-Send line until the host is ready to receive the 4161A's output. This is a reliable method.

Finally, the host can be programmed to accept the 4161A's output stream under interrupt control. This is a reliable method, but it requires extensive knowledge of both the host computer's hardware and assembly language.

The second problem is that once the 4161A's output stream starts, the host must be able to accept characters as fast as the 4161A can send them. This can be assured by selecting a very low baud rate. In order to allow data transfer at high baud rates, it may be necessary to code the host's input routine in assembly language.

When establishing serial communication, it is advisable to start out at a very low baud rate, such as 300 baud. Once the host's program is seen to be working, the baud rate can be raised as high as the host's software will allow.

It is possible to halt the 4161A's output in midstream by switching CLEAR-TO-SEND as previously described. However, the computer must be prepared to accept one or more final characters before transmission stops.

6. Try beginning with a very low baud rate, such as 300 baud. Once the host's program is seen to be working, the baud rate can be raised as high as the host's software will support.

   In order to allow data transfer at high baud rates, it may be necessary to code the host's input routine in assembly language. Whether it is written in a high-level language or Assembly, the host's program should accept the 4161A's flow of output data in a tight loop.

7. If communication problems occur, it might prove helpful to monitor the host computer's output line with an oscilloscope. One effective approach to doing this is to program the computer to transmit the letter "U" (alternate 1's and 0's) in a repeating loop. If the scope is adjusted to trigger about the ground level on a positive transition, and the X-axis display is calibrated so that the time per division equals one bit period, the serial data can be easily examined and decoded. Any RS232C parameter discrepancies can then be determined and corrected. The output of the 4161A can be similarly checked by turning ON the RS232C Local Echo function (FUNCTION F5) and observing the waveform at the serial port as the detector interface echoes the "U" back to the host computer.

If you find it necessary to change the default settings of the computer to make it work with the 4161A, be sure to bypass the computer's I/O re-initialization routines. These routines may restore the computer's default parameter values, causing messages to become garbled.

4.7 CIRCUIT DESCRIPTIONS

4.7A CLOCK GENERATOR

The clock generator consists of U34, a 16 MHz crystal-controlled TTL oscillator, whose output (pin 8) drives U31, which supplies the 8 MHz clock required by the microprocessor and the 4 MHz required by the GPIB hardware.

4.7B MICROPROCESSOR

The microprocessor, U40, is a Motorola MC68008, an eight-bit bus version of the 68000 family of microprocessors. The power-up logic for this microprocessor is supported by U45, which has delayed reset outputs to allow the +5 V to stabilize.

The interrupt logic is designed to support the auto-vector modes only, where the inputs are logically combined via U17, U46, and U48.

A single-chip 128K EPROM, U42, contains the firmware, along with an 8K by 8K RAM, U23.

Address decoding is generated by U43 and U38. Wait states are generated by U33's outputs, which are jumper-selected via J19 to control the DTACKLO (data transfer acknowledge) input of the microprocessor.

U44 is used during the power-up sequence to force the EPROM chip select to be mapped down to the lower eight bytes, which contains the power-up initialization vectors.

4.7C FRONT-PANEL LED DISPLAY DRIVER

U2 is an eight-bit latch driven by the microprocessor's data bus. Its outputs are emitter buffered by discrete transistors Q9 through Q13. The emitters of these transistors are used to drive the digit select of the LED's. Q9's emitter is the GPIB digit select. Q10 through Q13 are the digit drives (least significant to most significant) to the front-panel LED's. The digit drives are driven high to enable a digit.

U1 is an eight-bit latch driven by the data bus of the

microprocessor. Its outputs are again buffered Q1 through Q8 to supply the segment drives of the previously mentioned digits. They are driven low to turn on a segment.

4.7D A/D CONVERTER

Channel A and Channel B front-panel BNC's are fed via J2 to analog multiplexer U12. The output of the analog multiplexer is buffered by operational amplifier U4. The control for the multiplexer is the output of eight-bit latch U9, whose data inputs are on the microprocessor bus. U4's output (pin 6) is used to drive the input of A/D Converter U8. The converter is configured to run in a ± 10.24 V configuration to correspond to ± 2048 counts. The 12 bit output of this A/D Converter is multiplexed onto the microprocessor bus. Conversion takes place under microprocessor control only by strobing the chip select (pin 3 of U8).

4.7E FRONT-PANEL ANALOG INPUT

The front-panel analog input BNC is a totally independent input. Its function is solely to drive a zero-centered ± 1 mA panel meter. Full-scale input at this connector is ± 10 V into 10 kΩ.

4.7F TRIGGER INPUTS

Both the Channel A and Channel B Trigger Inputs are applied via J11 to the clock inputs of U16 (Interrupt flip-flops). The complementary outputs of these flip flops are used to supply the interrupt requests at the input of U17, which ultimately end up at microprocessor U40 to interrupt the program to commence a data-acquisition sequence.

The noninverting inputs of these flip-flops are used to drive open-collector buffer U18 in order to assert the rear-panel BUSY signal. In addition, BUSY is also asserted by the outputs of U18B and U18D. The input to U18B is a microprocessor controlled parallel output bit of U15, and is used to have ultimate control over the rear-panel BUSY function. The input of U18D's input comes from the A/D Converter U8's Status Line output. This Status Line also asserts the rear-panel BUSY output.

4.7G GPIB

The setup parameters for the GPIB, i.e., addresses, carriage return, and terminator, are set by a DIP switch assembly S1 on the main board. The outputs of these switches are read onto the microprocessor bus buffer U10. The switch settings are only read at power up.

A three-chip set (U24, U20, & U21) provide the GPIB interface functions. U24 is the Controller chip and U20 and U21 are the Bus Buffer chips.

4.7H RS232

The setup parameters for the RS232 port, number of stop bits, echo, parity, baud rate, and character length, are set by DIP switch assembly S2. The outputs of these switches are read onto the microprocessor bus buffer U11. The switch settings are only read at power up.

RS232 functions are supported by a dual UART, U15. In addition, this UART also supports parallel I/O functions used for other purposes. The U13 buffers to the RS232 connector are used to translate the RS232 to the TTL levels required by U15. Similarly the buffers in U14 are used to translate the U15 levels to the required RS232 levels.

Appendix A
GPIB Background

A.1 INTRODUCTION

It is NOT at all necessary to understand the material in the following paragraphs to use the GPIB interface. This text is provided as background information only.

A.2 GENERAL INTERFACE MANAGEMENT LINES

There are five general interface management lines, all active low, as follow:

ATN: (ATTENTION)

IFC: (INTERFACE CLEAR)

REN: (REMOTE ENABLED)

EOI: (END OR INDENTIFY), and

SRQ: (SERVICE REQUEST)

Although a programmer typically doesn't have to be concerned with the details of how the General Interface Management Lines work, a few comments are provided.

This instrument recognizes ATN and IFC as defined in the standard.

The 4161A does not use the REM line. Both the GPIB and RS232C ports are active at all times.

The Model 4161A accepts EOI as a terminator for all messages and asserts EOI with the last byte of each response.

A.3 REMOTE MESSAGES

IEEE 488 defines a large number of multi-line messages the computer can place on the bus when ATN is asserted. Of these, the Model 4161A recognizes TALK, UNTALK, LISTEN, UNLISTEN, DCL (DEVICE CLEAR), SPE (SERIAL POLL ENABLE), and SPD (SERIAL POLL DISABLE). As was the case for the General Interface Management Lines, the programmer normally does not have to concern himself with the details of how these messages function. Each is discussed in the following paragraphs:

1. **TALK, UNTALK, LISTEN, and UNLISTEN:** The TALK and LISTEN messages direct the Model 4161A to assume the talker and listener states respectively. Similarly the UNTALK and UNLISTEN commands direct the Model 4161A to assume the talker-idle and listener-idle states respectively.

2. **DEVICE CLEAR:** This message restores the default values of all parameters.

3. **SERIAL POLL ENABLE and SERIAL POLL DISABLE:** These messages control the serial polling process. The SERIAL POLL ENABLE message tells the Model 4161A to prepare to place its status byte (eight-bit binary) on the bus. This actually occurs when the TALK address is applied. The SERIAL POLL DISABLE message directs the 4161A to take its Serial Poll Status Byte off of the bus. The serial poll sequence is:

   a. The computer sends the SERIAL POLL ENABLE message (ATN asserted).

   b. The computer sends the TALK message (ATN asserted).

   c. The computer deasserts ATN.

   d. The Model 4161A sends the Serial Poll Status Byte to the computer.

   e. After reading the Serial Poll Status Byte, the computer reasserts ATN.

   f. The computer sends the UNTALK message (ATN asserted).

   g. The computer sends the SERIAL POLL DISABLE message (ATN asserted).

Note that the computer can do a serial poll at any time.

Appendix B
RS232C Background

B.1 INTRODUCTION

RS232C is an industry standard serial data communications interface. It is widely used for communications between digital devices such as computers, terminals, printers, and telephone links. RS232C is capable of bidirectional data transfer between digital devices.

Both devices must be equipped with RS232C interfaces for data transfer to take place. Virtually every modern computer has RS232C capability as standard or optional equipment. RS232C interfacing specifies data transfer as the flow of a stream of serial characters, usually ASCII encoded. ASCII (American Standard Code for Information Interchange) coding defines each character as a specific set of seven bits. For example, the "V" in the six character sequence, "-10.2V", is defined as the following seven bit binary number: "1010110". When transmitting information, the RS232C circuit translates the characters it receives into the proper bit sequences and then sequentially sends these encoded characters to the receiving device. The RS232C interface functions in a likewise manner when receiving information.

The RS232C Standard specifies many parameters which affect data transfer, including voltage (logic) levels, transfer rates, "handshaking", etc. However, some parameters have optional values that are user selectable. The user must be aware of these parameters, since they must be set the same on both devices for proper communication to take place.

The RS232C parameters which are normally user-selectable include baud rate, word length, parity, and stop-bit configuration. These four parameters must be selected to be exactly the same on both devices. On some equipment, certain parameters may not be altered. In these cases, the other device in the system must be configured to match the fixed parameters.

B.2 RS232C WORD CONSTRUCTION

Between transmissions, the transmit data line rests at the low voltage level. Each transmission begins with a START bit, the leading edge of which is marked by the transmit line being raised to the positive voltage level. The START bit is then followed by the seven data bits that define the character being sent. Bit 8, if transmitted, is set at logic 0 for ASCII characters.

"Negative true logic" is used for data. The RS232C standard specifies that a logic 1 is represented by a negative voltage between -3 V and -25 V, while a logic 0 is represented by a positive voltage in the range of +3 V to +25 V. In RS232C communication, the ASCII character code is used with the Most Significant Bit sent first. If the user elects to include a parity bit, it directly follows the data bits that define the character.

The parity bit is followed by either one or two STOP bits, as selected. The STOP bits are at the negative voltage level, returning the line to the quiescent state. If the user has selected "no parity", the parity bit time slot will be deleted, and the time slot immediately following the last data bit will be allocated to a STOP bit.

B.3 HISTORY AND PROBLEMS OF RS232C

The remainder of this appendix consists of general discussion of the history and problems of RS232C. An understanding of this material is in no way essential to using the Model 1461A's RS232C interface. It is simply provided as a source of additional background information for those who may be interested.

Serial data transmission is not new. Beginning with Morse Code and its use in telegraphy, the technology of serial data transmission has improved steadily to the point where today there is wide-spread reliable electronic data communications via serial coding. With the development of computers, the efficiency of serial data transmission in communications between humans and computers was quickly recognized. Serial data communications in computer applications were widely adopted. Initially, teletypewriters were almost exclusively used as combination terminal/display devices. With time, terminals and display devices advanced far beyond the teletypewriters used initially, but the serial I/O data handling characteristics of teletypewriters was maintained.

Serial communications reached a critical point when the distances involved became long. Clearly it was impractical to string wires between widely separated devices if it could be avoided. The ready solution was to make use of the existing telephone line system. Unfortunately, the telephone system was not designed with serial data transmission in mind, but rather for voice communications, with telephones serving as the I/O devices. The situation was additionally complicated by the inclusion in the telephone system of non-wire "connections" such as microwave data links. The probable chaos of allowing direct connections between the various serial I/O devices and telephone lines was not acceptable, and some sort of standard had to be established. RS232C became that standard.

RS232C defines the signal requirements for communications between a DTE (Data Terminal Equipment) device, typically a computer or a video terminal, and a DCE (Data Communications Equipment) device, such as a modem. This standard has served well and is the basis for almost all long-distance data transmission. However, its very success has led to its adoption in applications for which it was not intended. Today there are countless communications links that use the RS232C standard, even though the standard is not

intended to apply to them. Computers, printers, measuring instruments, etc. are frequently provided with "RS232C compatible" serial data interfaces. However, since many of these devices are neither terminals nor modems, they cannot, by definition, comply strictly with RS232C. Although users may expect that two pieces of equipment having RS232C ports should successfully communicate as soon as they are connected, it frequently happens that they do not. There are several possible problems which may prevent successful communications. One problem results from the direction of data flow. The standard specifies that data transmitted from a terminal to a modem should be routed via pin 2 of the terminal connector. Data received from the modem must enter the terminal via pin 3 of that same connector. When the device in question is neither a terminal nor a modem, the manufacturer must select which pin is used to output data and which to receive data. As a result, it sometimes happens that users have to cross-wire from pin 2 to pin 3 (via a "modem eliminator cable") to allow two "RS232C Standard" devices to communicate.

Handshake compatibility can be another serious impediment. "Handshaking" is the term which is used to describe the method that one digital device uses to tell another digital device that it is ready to receive information. Data may be lost if it is sent to a device that is not ready to receive it. RS232C has limited handshake requirements. The "RS232C Standard" is concerned solely with communication between terminals and modems. Therefore, the Request-to-Send and Clear-to-Send lines (pins 4 and 5 respectively) fulfill that requirement. These lines do not necessarily satisfy the handshake requirements for other equipment. According to the standard, the DTE device is supposed to assert Request-to-Send when it has data ready to be sent. Having asserted Request-to-Send, it is then supposed to wait for Clear-to-Send to be asserted by the DCE device. Once data transfer begins the DCE device is not allowed to drop the Clear-to-Send line low until the DTE drops Request-to-Send. Clearly there is a potential for the DCE device being asked to "take a longer drink" than it can handle. If the Clear-to-Send line is arbitrarily used as a "spigot" to turn the data flow On and Off (definitely NOT according to the standard), there is the question of determining just when it should switch. If it changes state when its buffer is full, it might stop in the middle of a character and garble the data. However, if it waits, the buffer will overflow and data will be lost. The problem is simply that RS232C was not designed for the job it is often asked to do, even though it can generally be made to work.

Another problem has its origin in the RS232C voltage levels. Because TTL (Transistor-Transistor Logic) is so widely used in modern electronic designs, it is all too easy to assume that the same TTL devices and levels apply everywhere. However, RS232C existed long before TTL, and its voltage levels differ from those of TTL. In RS232C, negative true logic is used for data communications; positive true logic is used for commands. The RS232C standard voltage levels are +3 V to +25 V for the positive logic level, and -3 V to -25 V for the negative logic level. Because the region between ±3 V is undefined, designers cannot use TTL devices in implementing RS232C communications.

When two digital devices are being set up for RS232C communication, users also must be sure that both pieces of equipment operate at the same data-transmission rate (baud rate), agree on parity, have the same number of stop bits, and use the same character code (usually ASCII). A mismatch in any one of these parameters will cause data to be garbled or lost.

Between data transmissions, the data lines idle at the negative logic level. As shown in Figure B-1, each transmission begins with a start bit, the leading edge of which is defined by the data line going positive (logic '0'). The start bit lasts for one bit interval, the duration of the bits as set by the baud rate, and acts as a signal to the receiving device to begin sequential sampling of the level on the line at the center of each bit's duration. Clearly, both devices must agree as to the number of bits to follow if they are to communicate successfully. The start bit is followed by the data bits (seven for ASCII) that define the character being transmitted.

The parity bit is sent next. RS232C provides for even parity, odd parity, or no parity (in which case there is no parity-bit time interval). However, both instruments have to be set the same with respect to the parity bit. The parity bit is then followed by either one or two stop bits. The line then comes back to the negative voltage level for the duration of the stop bits. It continues idling at the negative level until another start bit is transmitted, marking the beginning of new data. Most RS232C compatible devices use the ASCII code and provide for user selection and control of the parity bit, the number of stop bits, and the baud rate. Sometimes eight bits are used instead of seven, which is convenient for byte oriented I/O devices. Successful RS232C communications can usually be established; it's just not as straight forward as expected.

Serial interfacing is popular and reliable. However, when

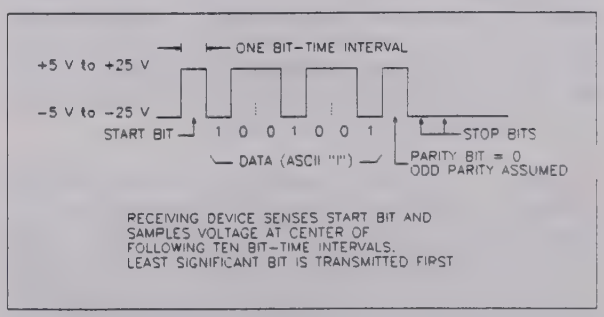

ONE BIT-TIME INTERVAL
+5 V to +25 V
-5 V to -25 V
START BIT
1 0 0 1 0 0 1
STOP BITS
DATA (ASCII "I")
PARITY BIT = 0
ODD PARITY ASSUMED
RECEIVING DEVICE SENSES START BIT AND
SAMPLES VOLTAGE AT CENTER OF
FOLLOWING TEN BIT-TIME INTERVALS.
LEAST SIGNIFICANT BIT IS TRANSMITTED FIRST

Figure B-1. TYPICAL RS232C DATA WORD

two pieces of equipment are to communicate serially, the communications link must be established with care and attention must be paid to detail. This is true even though both devices have serial ports which are implemented "in accordance with the RS232C Standard". Users who wish to receive more detailed information about the RS232C Standard can purchase a copy from:

Electronic Industries Association
Engineering Department
2001 I Street, N.W.
Washington, DC

Appendix C
4144 Interface Routine

INTERFACE ROUTINE: The following Static Interface Routine may be helpful in establishing computer control of the Model 4144. Moreover, it may be used as a component of more complex control programs.

```
#include <stdio.h>
#include <stdlib.h>

/* ------------------------------------------------------------------------
/
/   si4144.c
/   (C) Copyright 1989, EG&G Princeton Applied Research, Inc.
/
/   Written by: TLB          Version 1.00        16 February     1988
/   Worked on:  TLB          Version 1.01        19 May          1988
/
/   si4144.c is a static interface for the 4144 Digital Delay Generator.
/   Uses the National Instruments GPIB interface and interface control code.
/
make file (Microsoft C compiler V5.00, Microsoft Make Utility)
-------------------------------------------------------------
SI4144.OBJ:     SI4144.C
    CL -c -AS SI4144.C

SI4144.EXE:  SI4144.OBJ
    LINK SI4144+mcibS,SI4144,NUL;
-------------------------------------------------------------
(the file MCIBS.OBJ is supplied by National Instruments, Inc.)
/ ------------------------------------------------------------------------ */

#define FALSE           0
#define TRUE            1
#define NOT             !
#define CR              0x0d

typedef int boolean;

/* #include "gpib.h" */
/* = = = = = = = = = = = = = = = = = = = = = = = = = = = = =                */
/* these definitions were originally in an include file called "gpib.h"     */

extern  int     ibsta;  /* status word                                      */
extern  int     iberr;  /* GPIB error code                                  */
extern  int     ibcnt;  /* number of bytes sent                             */

/* GPIB Commands                                                           */
#define UNL  0x3f  /* GPIB unlisten command                                */
#define UNT  0x5f  /* GPIB untalk command                                  */
#define GTL  0x01  /* GPIB go to local                                     */
#define SDC  0x04  /* GPIB selected device clear                           */
#define PPC  0x05  /* GPIB parallel poll configure                         */
#define GET  0x08  /* GPIB group execute trigger                           */
#define TCT  0x09  /* GPIB take control                                    */
#define LLO  0x11  /* GPIB local lock out                                  */
#define DCL  0x14  /* GPIB device clear                                    */
#define PPU  0x15  /* GPIB parallel poll unconfigure                       */
#define SPE  0x18  /* GPIB serial poll enable                              */
#define SPD  0x19  /* GPIB serial poll disable                             */
#define PPE  0x60  /* GPIB parallel poll enable                            */
```

```
#define PPD  0x70  /* GPIB parallel poll disable                    */

/* GPIB status bit vector :                                         */
/*       global variable ibsta and wait mask                        */
#define ERR      0x8000  /* Error detected                         */
#define TIMO     0x4000  /* Timeout                                */
#define END      0x2000  /* EOI or EOS detected                    */
#define SRQI     0x1000  /* SRQ detected by CIC                    */
#define RQS      0x0800  /* Device needs service                   */
#define CMPL     0x0100  /* I/O completed                          */
#define LOK      0x0080  /* Local lockout state                    */
#define REM      0x0040  /* Remote state                           */
#define CIC      0x0020  /* Controller-in-Charge                   */
#define ATN      0x0010  /* Attention asserted                     */
#define TACS     0x0008  /* Talker active                          */
#define LACS     0x0004  /* Listener active                        */
#define DTAS     0x0002  /* Device trigger state                   */
#define DCAS     0x0001  /* Device clear state                     */

/* Error messages returned in global variable iberr                */
#define EDVR  0  /* DOS error                                      */
#define ECIC  1  /* Function requires GPIB board to be CIC         */
#define ENOL  2  /* Write function detected no Listeners           */
#define EADR  3  /* Interface board not addressed correctly        */
#define EARG  4  /* Invalid argument to function call              */
#define ESAC  5  /* Function requires GPIB board to be SAC         */
#define EABO  6  /* I/O operation aborted                          */
#define ENEB  7  /* Non-existent interface board                   */
#define EOIP 10  /* I/O operation started before previous          */
                 /* operation completed                            */
#define ECAP 11  /* No capability for intended operation           */
#define EFSO 12  /* File system operation error                    */
#define EBUS 14  /* Command error during device call               */
#define ESTB 15  /* Serial poll status byte lost                   */
#define ESRQ 16  /* SRQ remains asserted                           */

/* EOS mode bits                                                    */
#define BIN           0x1000  /* Eight bit compare                 */
#define XEOS_SETS_EOI 0x0800  /* Send EOI with EOS byte            */
#define REOS_STOPS    0x0400  /* Terminate read on EOS             */

/* Timeout values and meanings                                      */
#define TNONE    0   /* Infinite timeout (disabled)               */
#define T10us    1   /* Timeout of 10 us (ideal)                  */
#define T30us    2   /* Timeout of 30 us (ideal)                  */
#define T100us   3   /* Timeout of 100 us (ideal)                 */
#define T300us   4   /* Timeout of 300 us (ideal)                 */
#define T1ms     5   /* Timeout of 1 ms (ideal)                   */
#define T3ms     6   /* Timeout of 3 ms (ideal)                   */
#define T10ms    7   /* Timeout of 10 ms (ideal)                  */
#define T30ms    8   /* Timeout of 30 ms (ideal)                  */
#define T100ms   9   /* Timeout of 100 ms (ideal)                 */
#define T300ms  10   /* Timeout of 300 ms (ideal)                 */
#define T1s     11   /* Timeout of 1 s (ideal)                    */
#define T3s     12   /* Timeout of 3 s (ideal)                    */
#define T10s    13   /* Timeout of 10 s (ideal)                   */
#define T30s    14   /* Timeout of 30 s (ideal)                   */
#define T100s   15   /* Timeout of 100 s (ideal)                  */
#define T300s   16   /* Timeout of 300 s (ideal)                  */
#define T1000s  17   /* Timeout of 1000 s (maximum)               */
```

```
/* OR these with device addresses to address for Listen or Talk                */
#define   LISTEN              0x20
#define   TALK                0x40

/* Miscellaneous                                                               */
#define S    0x08    /* parallel poll sense bit                               */
#define LF   0x0a    /* ASCII linefeed character                              */

#define MIN_GPIB_ADDR 0
#define MAX_GPIB_ADDR 31

/* = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = */

#define ADDR_INST 2   /* default instrument GPIB address is 2                  */
#define ADDR_PC   0   /* PC's GPIB address is zero                             */

/* ------------------------------------------------------------------------
/   global data area
/ ------------------------------------------------------------------------ */

char       board_name[] = "GPIB0";       /* PC GPIB interface board 0          */

char       gpib_out[50];

/*  GPIB command strings */
unsigned char    talk_pc[10];
unsigned char    untalk[5];
unsigned char    ser_poll[10];
unsigned char    ser_poll_done[10];

int  board_id;                           /* 'handle' from board open call      */
int  address_instrument = ADDR_INST;
int  address_pc = ADDR_PC;

/* macros to build command strings (count in first byte) */
#define clears(string)  string[0] = 0
#define builds(string,command_byte) string[++string[0]] = command_byte

/* macro to simplify calling the ibcmd() function */
#define command(board,string)       ibcmd(board,&string[1],string[0])


/* ------------------------------------------------------------------------
/   code section
/ ------------------------------------------------------------------------ */

/* ------------------------------------------------------------------------
/   build_command_strings()
/
/   function:     loads GPIB bus command sequences into pre-fab strings
/
/   requires:     (void)
/   returns:      (void)
/ ------------------------------------------------------------------------ */

build_command_strings()
{
    clears(talk_pc);
    builds(talk_pc, UNL);
    builds(talk_pc, UNT);
    builds(talk_pc, (TALK | address_pc));
```

```
    builds(talk_pc, (LISTEN | address_instrument));

    clears(untalk);
    builds(untalk, UNL);
    builds(untalk, UNT);

    clears(ser_poll);
    builds(ser_poll, UNL);
    builds(ser_poll, UNT);
    builds(ser_poll, SPE);
    builds(ser_poll, (LISTEN | address_pc));
    builds(ser_poll, (TALK | address_instrument));

    clears(ser_poll_done);
    builds(ser_poll_done, UNL);
    builds(ser_poll_done, UNT);
    builds(ser_poll_done, SPD);
}

/* ------------------------------------------------------------------
/   init_gpib()
/
/   function: called to open and initialize the PC's GPIB interface.
/
/   requires:     (char *) boardname - text string describing the board,
/                       probably "GPIB0" or "GPIB1".
/   returns:      (boolean) success flag - TRUE if board opened OK.
/  ------------------------------------------------------------------ */

boolean init_gpib(boardname)
char *boardname;
{
    if ((board_id = ibfind(boardname)) < 0 )
        return(FALSE);
    else
    {
        ibsic(board_id);                        /* interface clear */
        ibtmo(board_id, T3s);                   /* set 3 second timeout */
        return(TRUE);
    }
}

/* ------------------------------------------------------------------
/   shutdown_gpib()
/
/   function: closes down the GPIB interface when all done.
/
/   requires:     (void)
/   returns:      (void)
/  ------------------------------------------------------------------ */

void shutdown_gpib()
{
    ibsic(board_id);
    ibonl(board_id, FALSE);
}

/* ------------------------------------------------------------------
/   puts_gpib()
/
/   function: called to write a string to the instrument.
```

```
/
/   requires:     (char *) string - a string buffer to send to the instrument.
/                       (int *) count - the number of bytes from the
/                       buffer to send to the instrument.
/   returns:      (unsigned int) error status - from the global variable ibsta.
/                       (int *) count - will be set with the actual number
/                       of bytes written to the instrument.
/ ---------------------------------------------------------------------- */

unsigned int puts_gpib(string,count)
char *string;
int *count;
{
    command(board_id, talk_pc);
    ibwrt(board_id, string, *count);
    *count = ibcnt;
    command(board_id, untalk);
    return (ibsta);
}

/* ---------------------------------------------------------------------
/   serial_poll_gpib()
/
/   function: gets the serial poll byte from the instrument.
/
/   requires:     (void)
/
/   returns:      (unsigned char) the serial poll byte.
/
/ ---------------------------------------------------------------------- */

unsigned char serial_poll_gpib()
{
    unsigned char poll_byte;
    unsigned char reply[3];
    unsigned int standard_timeout = T10s;

    command(board_id, ser_poll);        /* address instrument, enable serial poll */
    ibtmo(board_id, T300ms);            /* shorten timeout period                 */
    if ( NOT (ibsta & ERR))             /* save old timeout value (in iberr       */
        standard_timeout = iberr;       /* unless there was an error              */
    ibrd(board_id, reply, 1);           /* read one serial poll byte              */
    if (ibcnt != 1)
    {
        printf("\nERROR on serial poll, returned %d bytes - s.b. 1",ibcnt);
        poll_byte = 0;
    }
    else
        poll_byte = reply[0];
    ibtmo(board_id, standard_timeout);  /* restore previous timeout value         */
    command(board_id, ser_poll_done);   /* disable serial poll                    */
    return (poll_byte);
}

/* ---------------------------------------------------------------------
/   analyze_serial_poll()
/
/   function: displays the bits set in the poll byte.
/
/   requires:     (unsigned char) poll_byte - gathered from the instrument via
/                       the serial_poll_gpib() call.
```

```
/
/   returns:      (void)
/
/ ------------------------------------------------------------------- */

void analyze_serial_poll(poll_byte)
register unsigned char poll_byte;
{
    if (poll_byte)
    {
        if (poll_byte & 0x80)          printf("7  ");
        if (poll_byte & 0x40)          printf("6  ");
        if (poll_byte & 0x20)          printf("5  ");
        if (poll_byte & 0x10)          printf("4  ");
        if (poll_byte & 0x08)          printf("3  ");
        if (poll_byte & 0x04)          printf("<PEN LIFT (2)>  ");
        if (poll_byte & 0x02)          printf("<PARAM ERR (1)>  ");
        if (poll_byte & 0x01)          printf("<SCANNING (0)>  ");
    }
    else
    {
        printf("(no bits set)");
    }
}

void show_serial_poll()
{
    printf("\n>>  Serial Poll  >>  ");
    analyze_serial_poll( serial_poll_gpib() );
}

/* ------------------------------------------------------------------
/   void process_argument(arg)
/
/
/   function: does the appropriate processing for each of the command line
/         arguments, which are fed to this function one at a time.
/             (only one argument for this program.)
/
/   requires:     (char *) arg - an argument parsed from the command line
/   returns:      (void)
/ ------------------------------------------------------------------- */

void process_argument(arg)
char *arg;
{
    if (arg[0] == '-' || arg[0] == '/')     /* switches begin with '-' or '/'          */
    {
        switch (arg[1])
        {
            case 'a':                       /* -a instrument's GPIB address            */
                address_instrument = atoi(&arg[2]);
                if (address_instrument < MIN_GPIB_ADDR ||
                    address_instrument > MAX_GPIB_ADDR ||
                    address_instrument == address_pc)
                {
                    address_instrument = ADDR_INST;
                    printf("\nInstrument GPIB address out of range or same as PC\n");
                }
                break;
```

```
                default:                    /* -? what was that?                */
                    printf("\nArgument error: >>%s<<", arg);
            }
        }
        else
        {
            /* no non-switch options for this program */
        }
}

/* ------------------------------------------------------------------
/
/   int fractional_to_int(str, digits)
/
/   function:   converts a fraction string to an integer value.
/               used instead of atoi() because we must have strict
/               control of how many fractional digits are significant.
/
/   requires:   (char *) str - contains a string of digits
/               (int) digits - the number of fraction digits allowed
/   returns:    (int) - the fraction value
/  ------------------------------------------------------------------ */

int fractional_to_int(str, digits)
char *  str;
int     digits;
{
    int     frac_value = 0;

    if (digits > 0)
    {
        int     i;
        int     digit_value;
        int     factor;
        int     end_of_str_found = FALSE;

        for (i=0; i < digits; i++)
        {
            if (frac_value)
                frac_value *= 10;
            if (NOT end_of_str_found)
            {
                if (str[i] == '\0')
                    end_of_str_found = TRUE;
                else
                {
                    digit_value = (str[i] - '0');
                    if (digit_value >= 0 && digit_value <= 9)
                        frac_value += digit_value;
                }
            }
        }
    }
    return(frac_value);
}

/* ------------------------------------------------------------------
/
/   char * get_fixed_point_string(whole,frac)
/
/   function:   converts a number string in '0000.00' format to
```

```
/                       '000000' format for the 4144, which has implicit
/                       decimal point positions, different for almost every
/                       command...
/
/   requires:       (char *) in_string - the number string
/                       (int) whole - the acceptable number of whole digits
/                       (int) frac - the acceptable number of fraction digits
/   returns:        (char *) - fixed point string
/  ------------------------------------------------------------------------ */

char * convert_to_fixed_point_string(in_string,whole, frac)
char * in_string;
int  whole;
int  frac;
{
    static char         fix_str[20];
    char                format_str[20];
    long                whole_part;
    int                     fractional_part = 0;
    int                     i;

    whole_part = atol(in_string);

    if (frac > 0)
    {
        int                 numstrlen;
        int                 dot = 0;

        if (numstrlen = strlen(in_string))
        {
            for (i=0; i < numstrlen; i++)       /* search for decimal point */
            {
                if (in_string[i] == '.')
                {
                    dot = i;
                    break;
                }
            }
            if (dot > 0 && dot < numstrlen)     /* if found, get fractional part */
            {
                fractional_part = fractional_to_int(&in_string[dot+1],frac);
            }
        }                                   /* create appropriate format string */
        sprintf(format_str, "%%0%dld%%0%dd", whole, frac);
        sprintf(fix_str, format_str, whole_part, fractional_part);
    }
    else
    {                                   /* create appropriate format string */
        sprintf(format_str, "%%0%dld", whole);
        sprintf(fix_str, format_str, whole_part);
    }
    return(fix_str);
}

/* ------------------------------------------------------------------------
/
/   boolean get_command(cmd_str)
/
/   function:       shows a menu and gets a command from the user.
/                       commands with parameters are in the format
/                       'C0000.00' (fractional part optional), where the
```

```
/                       the command character is immediately followed by
/                       the parameter;  otherwise the command character is
/                       entered alone.  the parameter is converted into
/                       a format the 4144 can understand, and the command
/                       is formatted to send out on the GPIB bus.
/
/   requires:       (char *) command_string - gets filled with the 4144 command
/   returns:        (boolean) - flag indicating a valid command
/  ------------------------------------------------------------------- */

boolean get_command(command_string)
char *      command_string;
{
    char            cmd_char;
    char            input_string[20];
    boolean         get_param_flag;
    boolean         done_flag;
    boolean         exit_flag;
    int             whole_digits;
    int             frac_digits;

    do
    {
        show_serial_poll();

        get_param_flag = TRUE;
        exit_flag = FALSE;
        done_flag = TRUE;

        printf("\n\n------- 4144 Command Options -------\n");
        printf("\n\t(A) Channel A Delay");
        printf("\n\t(B) Channel B Delay");
        printf("\n\t(E) Trigger Rate");
        printf("\n\t(F) Toggle Pen Lift In/Out");
        printf("\n\t(G) Scan Delay Step");
        printf("\n\t(H) Triggers/Step");
        printf("\n\t(I) Steps/Scan");
        printf("\n\t(J) Initiate Continuous Scanning");
        printf("\n\t(K) Initiate Single Scan");
        printf("\n\t(x) Exit Program");
        printf("\n\n\tCommand: ");

        gets(input_string);
        cmd_char = toupper(input_string[0]);
        switch (cmd_char)
        {
            case 'A':
            case 'B':
                whole_digits = 8;
                frac_digits = 2;
                break;
            case 'E':
                whole_digits = 7;
                frac_digits = 3;
                break;
            case 'F':
                /* get_param_flag = FALSE; */       /* flag should be FALSE, but */
                                                    /* 4144 seems to want params?!? */
                    input_string[1] = '0';          /* so, time to kludge... */
                    input_string[2] = '\0';
                    whole_digits = 1;
```

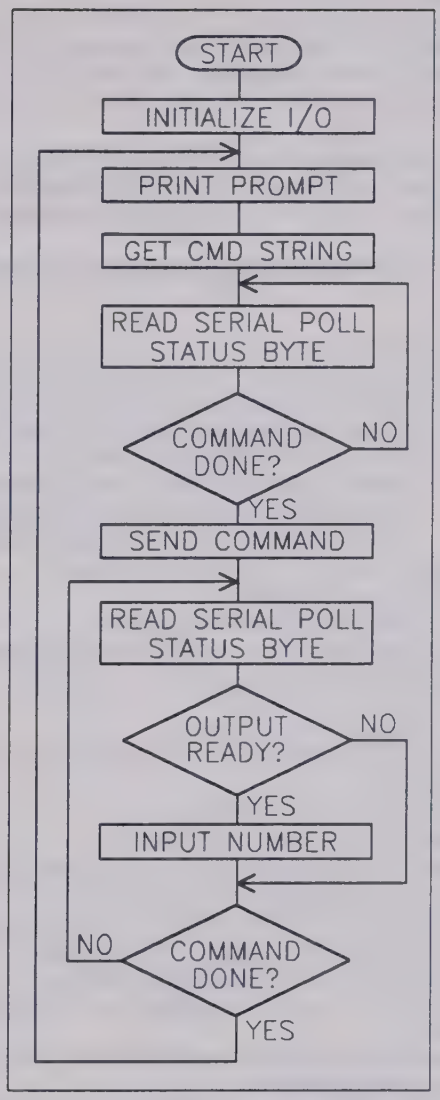
```
                        frac_digits = 0;                /* end_of_kludge */
                    break;
                case 'G':
                    whole_digits = 8;
                    frac_digits = 2;
                    break;
                case 'H':
                    whole_digits = 5;
                    frac_digits = 0;
                    break;
                case 'I':
                    whole_digits = 4;
                    frac_digits = 0;
                    break;
                case 'J':
                case 'K':
                    get_param_flag = FALSE;
                    break;
                case 'X':
                    exit_flag = TRUE;
                    break;
                default:
                    done_flag = FALSE;
                    break;
            }
        }
        while ( NOT done_flag );

        if ( NOT exit_flag )
        {
            if (get_param_flag)
            {
                sprintf(command_string, "%c\n%c%s\n", cmd_char, cmd_char,
                    convert_to_fixed_point_string(&input_string[1],
                    whole_digits, frac_digits) );
            }
            else
            {
                sprintf(command_string, "%c\n", cmd_char);
            }
        }
        return( NOT exit_flag);
}

/* ---------------------------------------------------------------------
/   main()
/
/   function:   static interface program for the 4144 Digital Delay Generator
/
/   requires:   (void)
/   returns:    (int) ERRORLEVEL - DOS return code (0 - ok, 1 - no GPIB board).
/  ------------------------------------------------------------------- */

int main(argc, argv)
int argc;
char *argv[];
{
    int     i;
    int     command_len;

    if (argc > 1)
```

```
    {
        for (i=1; i < argc; i++)
            process_argument(argv[i]);
    }
    if (init_gpib(board_name))
    {
        build_command_strings();

        printf("\nConnected to instrument...");
        while (get_command(gpib_out))
        {
            command_len = strlen(gpib_out);
            puts_gpib(gpib_out, &command_len);        /* send to instrument */
        }

        shutdown_gpib();
        return(0);                                    /* DOS ERRORLEVEL 0 (all ok) */
    }
    else
    {
        printf("\nERROR - cannot open the GPIB interface!");
        return(1);                                    /* DOS ERRORLEVEL 1 (no go) */
    }
}
```

Appendix D
4161A Static Interface Routine

D.1 INTRODUCTION

The Static Interface Routine is an example of a program suitable for use with the Model 4161A. A typical flowchart for this routine, which has been coded for several different computers, is shown in Figure D-1.



Note: Display of output values and error messages not indicated.

Figure D-1. STATIC INTERFACE ROUTINE FLOWCHART

D.2 SIMPLE STATIC INTERFACE PROGRAMS

D.2A IBM PC/XT/AT

This program is designed to run on IBM PC/XT/AT or compatible computers using the National Instruments GPIB-PC Model PC2A IEEE-488 GPIB Interface Card. Note that program lines 100, 105, and 110 are from the National Instruments BASICA declaration file DECL.BAS, and that the file BIB.M is also supplied by National Instruments. Each software revision has its own version of these files.

The interface needs to be configured by means of the National Instruments program IBCONF.EXE. It is advisable, at least during program development, to disable DMA and interrupts when configuring the interface, by setting 'DMA Channel' to 'NONE' and 'Interrupt jumper setting' to 'NONE'.

It may be worth noting that this program can send command files, but the commands in the files must not generate responses.

```
10  'GPSEND2 24-OCT-88 ..... for versions 29104, 8921
100 CLEAR, 60000! :IBINIT1=60000!:IBINIT2=IBINIT1+3:BLOAD"BIB.M",IBINIT1
105 CALL IBINIT1(IBFIND,IBTRG,IBCLR,IBPCT,IBSIC,IBLOC,IBPPC,IBBNA,
          IBONL,IBRSC,IBSRE,IBRSV,IBPAD,IBSAD,IBIST,IBDMA,IBEOS,IBTMO,
          IBEOT,IBRDF,IBWRTF,IBTRAP)
110 CALL IBINIT2(IBGTS,IBCAC,IBWAIT,IPPOKE,IBWRT,IBWRTA,IBCMD,IBCMDA,
          IBRD,IBRDA,IBSTOP,IBRPP,IBRSP,IBDIAG,IBXTRC,IBRDI,IBWRTI,
          IBRDIA,IBWRTIA,IBSTA%,IBERR%,IBCNT%)
115 '....assign access code to interface board.....
120 BDNAME$="GPIB0"
125 CALL IBFIND(BDNAME$,GPIB0%)
130 IF GPIB0%<0 THEN PRINT "board assignment error":END
135 'send INTERFACE CLEAR.....
140 CALL IBSIC(GPIB0%)
145 '....assign access code to device.....
150 CLS: PRINT"DEVICE MUST BE SET TO CR TERMINATOR"
155 SUCCESS%=0
160 WHILE (NOT SUCCESS%)
165   INPUT "BUS ADDRESS ";A%
170   DEVNAME$="DEV"+RIGHT$(STR$(A$),LEN(STR$(A%))-1)
175   CALL IBFIND(DEVNAME$,DEV%)
180   IF DEV%<0 THEN PRINT "device assignment error":END
185   A$=CHR$(13): GOSUB 395'write A$
190   IF IBSTA%>0 THEN SUCCESS%=-1
195   IF (IBSTA%<0 AND IBERR%=2) THEN BEEP: PRINT "NO DEVICE AT THAT ADDRESS"
200 WEND
205 '....send SELECTED DEVICE CLEAR, wait 1 second....
210 CALL IBCLR(DEV%)
215 TT=TIMER: WHILE TIMER<TT+1: WEND
220 '....set timeout to 1 second.....
225 V%=11: CALL IBTMO(DEV%,V%)
230 '....set status print.....
235 R$=""
240 WHILE (R$<>"y" AND R$<>"Y" AND R$<>"n" AND R$<>"N")
245   PRINT"Display status byte (y or n) ?"
250   R$=INPUT$(1)
255 WEND
260 IF R$="Y" OR R$="y" THEN DS%=-1 ELSE DS%=0
265 '....main loop.....
270 S%=0
275 WHILE 1
280   WHILE (S% AND 1)=0
285     GOSUB 430'serial poll
290   WEND
295   IF (S% AND 4) THEN PRINT "parameter error"
300   IF (S% AND 2) THEN PRINT "invalid command"
```

```
305   CCFLAG%=0
310   INPUT "GPIB command / ff (filesend) / ee (exit) ";A$
315   IF A$="ee" OR A$="EE" THEN END
320   IF A$="ff" OR A$"FF" THEN GOSUB 440: CCFLAG%=-1
325   IF A$="*" THEN GOSUB 500: CCFLAG%=-1
330   WHILE (NOT CCFLAG%)
335     A$=A$+CHR$(13)
340     GOSUB 395'write A$ to bus
345     S%=0
350     WHILE (S% AND 1)=0
355       GOSUB 430'serial poll
360       IF DS% THEN PRINT "S%= ";S%
365       IF (S% AND 128) THEN GOSUB 410: PRINT B$'read B$ from bus and print
370     WEND
375     CCFLAG%=-1
380   WEND
385 WEND
390 '....end of main loop.....................
395 '....write string to bus .....................
400 CALL IBWRT(DEV%,A$)
405 RETURN
410 '....read string from bus.....................
415 B$=SPACE$(20)
420 CALL IBRD(DEV%,B$)
425 RETURN
430 '....serial poll.....................
435 CALL IBRSP(DEV%,S%):RETURN
440 '....send command file.....................
445 INPUT "filespec ";N$
450 OPEN N$ FOR INPUT AS #1
455 WHILE (NOT EOF(1))
460   LINE INPUT #1,A$
465   PRINT A$: A$=A$+CHR$(13)
470   GOSUB 395'write A$ to bus
475 WEND
480 CLOSE #1
485 A$=CHR$(13)
490 GOSUB 395
495 RETURN
500 '....speedmode routine.....................
505 INPUT "How many reads ";NREAD%
510 FOR J%=1 TO NREAD%
515   GOSUB 395'write A$ to bus
520   GOSUB 410'read B$ from bus
525   PRINT B$
530 NEXT J%
535 PRINT
540 A$=CHR$(13)
545 GOSUB 395'write A$ to bus
550 RETURN
```

C.2B HEWLETT PACKARD 85

```
10 ! HP85 SIMPLE STATIC INTERFACE
20 RESET 7
30 SET TIMEOUT 7;5000
40 DIM A(100)

100 CLEAR
110 DISP "ENTER DEVICE GPIB ADDRESS: ";
115 INPUT A9
120 A9=A9+700
125 CLEAR A9 @ WAIT 1000
130 DISP "DEVICE MUST BE SET FOR"
140 DISP "LINEFEED TERMINATOR"
150 DISP ""
160 C$="DD 10" @ GOSUB 1000 ! LINEFEED IS DELIMITER

200 ! MAIN LOOP
210 DISP "ENTER COMMAND: ";
220 INPUT C$
230 GOSUB 1200 ! SEND COMMAND AND GET RESPONSES IF ANY
240 IF BIT (S9,1)=1 THEN DISP "COMMAND ERROR"
250 IF N=0 THEN 290
260 FOR I=1 TO N
270 DISP "RESPONSE#";I;" = ";A(I)
280 NEXT I
290 GOTO 200

1000 ! OUTPUT COMMAND
1010 GOSUB 1400 ! DO SERIAL POLL
1020 IF BIT (S9,0)=0 THEN 1010 ! WAIT FOR PREVIOUS CMD DONE
1030 OUTPUT A9 USING "K"; C$
1040 RETURN

1100 ! GET RESPONSES
1110 N=0
1120 GOSUB 1400 ! CHECK CMD DONE AND OUTPUT READY BITS
1130 IF BIT (S9,7)=0 THEN 1150
1140 N=N+1 @ ENTER A9;A(N)
1150 IF BIT (S9,0)=0 THEN 1120
1160 RETURN

1200 ! DEVICE DRIVER
1210 GOSUB 1000
1220 GOSUB 1100
1230 RETURN

1400 ! DO A SERIAL POLL
1410 S9=SPOLL (A9)
1420 RETURN
```

Appendix E
Pinouts

E.1 IEEE-488 GPIB PINOUT

Both the 4161A and the 4144 have a GPIB Interface.
Note that the GPIB signal and control lines are all active low.

PIN #	FUNCTION
1	DIO1
2	DIO2
3	DIO3
4	DIO4
5	EOI
6	DAV
7	NRFD
8	NDAC
9	IFC
10	SRQ
11	ATN
12	SHIELD
13	DIO5
14	DIO6
15	DIO7
16	DIO8
17	REN
18	GND 6
19	GND 7
20	GND 8
21	GND 9
22	GND 10
23	GND 11
24	LOGIC GROUND

Table E-1. GPIB PINOUT

E.2 NIM CONNECTOR
The power and ground connections are made via this connector, as follows.

PIN	USE
11	-6 V
17	-12 V
28	+24 V
29	-24 V
34	GROUND
16	+12 V
10	+6 V

Table E-2. NIM PINOUT

E.3 4161A RS232C CONNECTOR PINOUT
Note: The 4161A's RS232C port is configured as a DTE.

PIN	FUNCTION	COMMENT
1	Earth Ground	Ties the chassis of the Model 4161A to that of the Computer.
2	Transmit Data	The Model 4161A transmits on this line. It must connect to the computer connector pin that receives serial data.
3	Receive Data	The Model 4161A receives data on this line. It must connect to the computer connector pin that transmits serial data.
4	Request to Send	This line is permanently asserted in the Model 4161A, that is, it is always at a positive logic level (+10 V). As a result, the computer continuously receives the message that the Model 4161A is ready to receive a command.
5	Clear to Send	The computer controls the Model 4161A via this line. To enable the Model 4161A to transmit, the line is placed at the positive logic level (+3 V to +12 V). To hold off transmission by the Model 4161A, the line must be at the negative logic level (-3 V to -12 V). Once this line goes positive, data transfer, if initiated by an appropriate command, will proceed rapidly. This line is returned to +12 V via an internal resistor so that, if left unconnected, the 4161A is allowed unimpeded transmission.
6	Unused	
7	Logic Ground	Data signal levels should be with reference to logic ground. The logic ground line of the Model 4161A should interconnect with the logic ground line of the computer.

8 to 25 Unused

Note: The voltage levels defined in the RS232C Standard are:

DATA: Logic 0 is +3 V to +25 V.
Logic 1 is -3 V to -25 V.

CONTROL: Asserted is +3 V to +25 V.
Unasserted is -3 V to -25 V.

One might infer from these standard levels that the Model 4161A's RS232C output levels will be on the order of ±25 V. This is not the case. The actual output levels are approximately ±12 V, and the input levels must be less than ±15 V (higher levels will damage the 4161A).

Table E-3. RS232 CONNECTOR PINOUT

Appendix F
FCC Notice

THIS EQUIPMENT GENERATES, USES, AND CAN RADIATE RADIO-FREQUENCY ENERGY AND, IF NOT INSTALLED AND USED IN ACCORDANCE WITH THIS MANUAL, MAY CAUSE INTERFERENCE TO RADIO COMMUNICATIONS. AS TEMPORARILY PERMITTED BY REGULATION, OPERATION OF THIS EQUIPMENT IN A RESIDENTIAL AREA IS LIKELY TO CAUSE INTERFERENCE, IN WHICH CASE THE USER AT HIS OWN FACILITY WILL BE REQUIRED TO TAKE WHATEVER MEASURES MAY BE REQUIRED TO CORRECT THE INTERFERENCE.

Appendix G
ASCII Character Codes

BINARY	DECIMAL	HEX	CHARACTER	WHAT TO TYPE
0000000	0	00	NULL	CONTROL @
0000001	1	01	SOH	CONTROL A
0000010	2	02	STX	CONTROL B
0000011	3	03	ETX	CONTROL C
0000100	4	04	ET	CONTROL D
0000101	5	05	ENQ	CONTROL E
0000110	6	06	ACK	CONTROL F
0000111	7	07	BEL	CONTROL G
0001000	8	08	BS	CONTROL H
0001001	9	09	HT	CONTROL I
0001010	10	0A	LF	CONTROL J
0001011	11	0B	VT	CONTROL K
0001100	12	0C	FF	CONTROL L
0001101	13	0D	CR	CONTROL M
0001110	14	0E	SO	CONTROL N
0001111	15	0F	SI	CONTROL O
0010000	16	10	DLE	CONTROL P
0010001	17	11	DC1	CONTROL Q
0010010	18	12	DC2	CONTROL R
0010011	19	13	DC3	CONTROL S
0010100	20	14	DC4	CONTROL T
0010101	21	15	NAK	CONTROL U
0010110	22	16	SYN	CONTROL V
0010111	23	17	ETB	CONTROL W
0011000	24	18	CAN	CONTROL X
0011001	25	19	EM	CONTROL Y
0011010	26	1A	SUB	CONTROL Z
0011011	27	1B	ESCAPE	ESC
0011100	28	1C	FS	CONTROL \
0011101	29	1D	GS	CONTROL]
0011110	30	1E	RS	CONTROL ^
0011111	31	1F	US	CONTROL _
0100000	32	20	SPACE	SPACE
0100001	33	21	!	!
0100010	34	22	"	"
0100011	35	23	#	#
0100100	36	24	$	$
0100101	37	25	%	%
0100110	38	26	&	&
0100111	39	27	'	'
0101000	40	28	(	(
0101001	41	29	)	)
0101010	42	2A	*	*
0101011	43	2B	+	+
0101100	44	2C	,	,
0101101	45	2D	-	-
0101110	46	2E	.	.
0101111	47	2F	/	/
0110000	48	30	0	0
0110001	49	31	1	1
0110010	50	32	2	2
0110011	51	33	3	3
0110100	52	34	4	4
0110101	53	35	5	5
0110110	54	36	6	6

0110111	55	37	7	7
0111000	56	38	8	8
0111001	57	39	9	9
0111010	58	3A	:	:
0111011	59	3B	;	;
0111100	60	3C	<	<
0111101	61	3D	=	=
0111110	62	3E	>	>
0111111	63	3F	?	?
1000000	64	40	@	@
1000001	65	41	A	A
1000010	66	42	B	B
1000011	67	43	C	C
1000100	68	44	D	D
1000101	69	45	E	E
1000110	70	46	F	F
1000111	71	47	G	G
1001000	72	48	H	H
1001001	73	49	I	I
1001010	74	4A	J	J
1001011	75	4B	K	K
1001100	76	4C	L	L
1001101	77	4D	M	M
1001110	78	4E	N	N
1001111	79	4F	O	O
1010000	80	50	P	P
1010001	81	51	Q	Q
1010010	82	52	R	R
1010011	83	53	S	S
1010100	84	54	T	T
1010101	85	55	U	U
1010110	86	56	V	V
1010111	87	57	W	W
1011000	88	58	X	X
1011001	89	59	Y	Y
1011010	90	5A	Z	Z
1011011	91	5B	[	[
1011100	92	5C	\	\
1011101	93	5D	]	]
1011110	94	5E	^	^
1011111	95	5F		
1100000	96	60	‘	‘
1100001	97	61	a	a
1100010	98	62	b	b
1100011	99	63	c	c
1100100	100	64	d	d
1100101	101	65	e	e
1100110	102	66	f	f
1100111	103	67	g	g
1101000	104	68	h	h
1101001	105	69	i	i
1101010	106	6A	j	j
1101011	107	6B	k	k
1101100	108	6C	l	l
1101101	109	6D	m	m
1101110	110	6E	n	n
1101111	111	6F	o	o
1110000	112	70	p	p
1110001	113	71	q	q
1110010	114	72	r	r
1110011	115	73	s	s

1110100	116	74	t	t
1110101	117	75	u	u
1110110	118	76	v	v
1110111	119	77	w	w
1111000	120	78	x	x
1111001	121	79	y	y
1111010	122	7A	z	z
1111011	123	7B	{	{
1111100	124	7C	|	|
1111101	125	7D	}	}
1111110	126	7E	~	~
1111111	127	7F	RUBOUT	

CONTROL CHARACTER ALTERNATE CODE NAMES

CHARACTER	MEANING
NUL	NULL, CTRL SHIFT P, TAPE LEADER
SOH	START OF HEADER, SOM
STX	START OF TEXT, EOA
ETX	END OF TEXT, EOM
EOT	END OF TRANSMISSION, END
ENQ	ENQUIRY, WRU, WHO ARE YOU
ACK	ACKNOWLEDGE, RU, ARE YOU
BEL	BELL
BS	BACKSPACE, FE0
HT	HORIZONTAL TAB, TAB
LF	LINE FEED, NEW LINE, NL
VT	VERTICAL TAB, VTAB
FF	FORM FEED, FORM, PAGE
CR	CARRIAGE RETURN, EOL
SO	SHIFT OUT, RED SHIFT
SI	SHIFT IN, BLACK SHIFT
DLE	DATA LINK ESCAPE, DC0
DC1	XON, READER ON
DC2	TAPE, PUNCH ON
DC3	XOFF, READER OFF
DC4	TAPE, PUNCH OFF
NAK	NEGATIVE ACKNOWLEDGE, ERR
SYN	SYNCHRONOUS IDLE, SYNC
ETB	END OF TEXT BUFFER, LEM
CAN	CANCEL, CANCL
EM	END OF MEDIUM
SUB	SUBSTITUTE
ESC	ESCAPE, PREFIX
FS	FILE SEPARATOR
GS	GROUP SEPARATOR
RS	RECORD SEPARATOR
US	UNIT SEPARATOR

Appendix H
Schematics

H.1 4121B SCHEMATICS

Signal Board Front Layout Diagram 68

Signal Board Back Layout Diagram 69

Signal Board (19976-D-SD; sheet 1 of 6) 70

Signal Board (19976-D-SD; sheet 2 of 6) 71

Signal Board (19976-D-SD; sheet 3 of 6) 72

Signal Board (19976-D-SD; sheet 4 of 6) 73

Signal Board (19976-D-SD; sheet 5 of 6) 74

Signal Board (19976-D-SD; sheet 6 of 6) 75

Delay Board Layout Diagram 76

Delay Board Schematic (19983-D-SD) 77

H.2 4161A SCHEMATICS

Main Board Layout Diagram 78

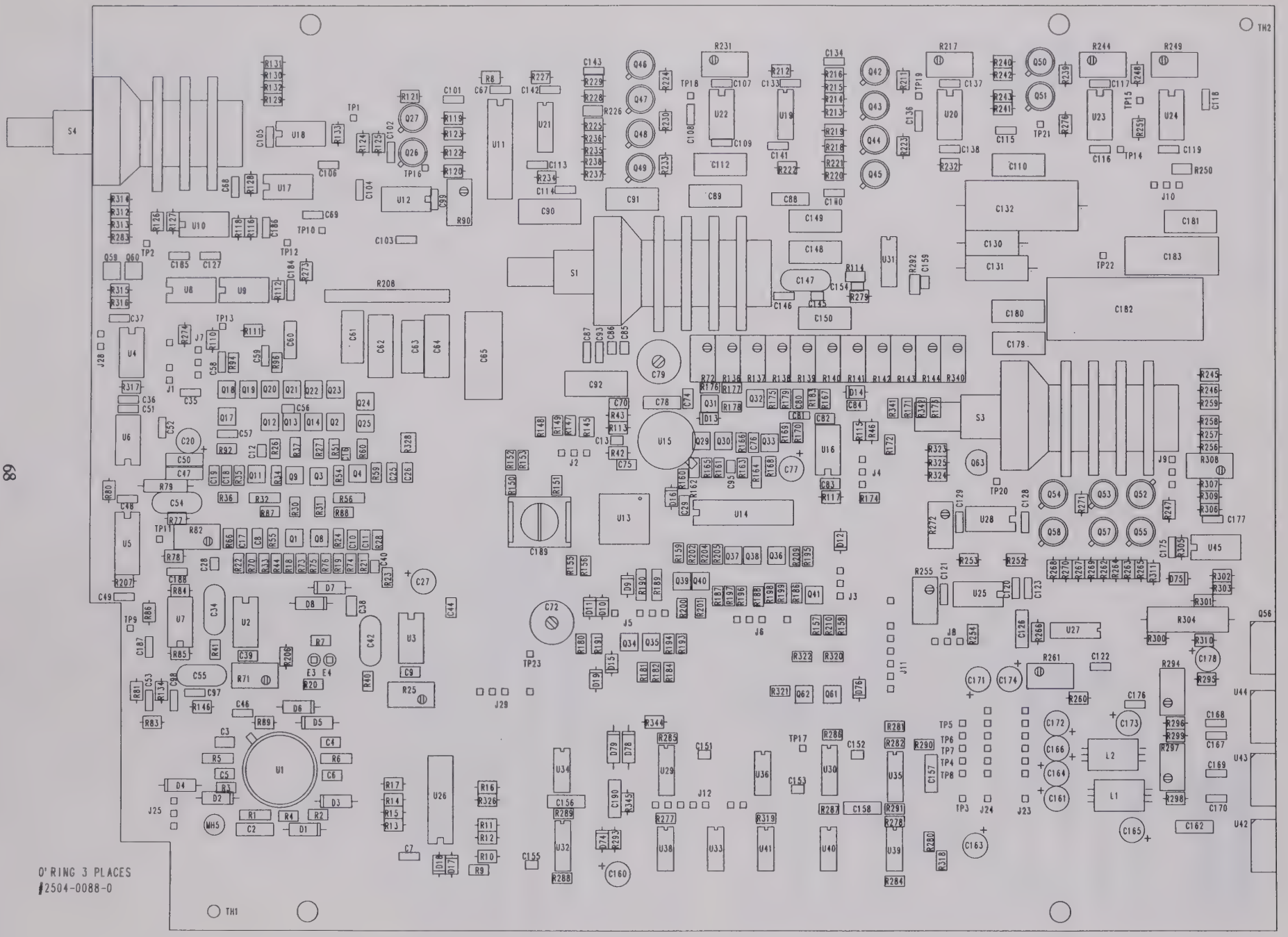
Main Board Schematic (19911-D-SD, sheet 1 of 3) 79

Main Board Schematic (19911-D-SD, sheet 2 of 3) 80

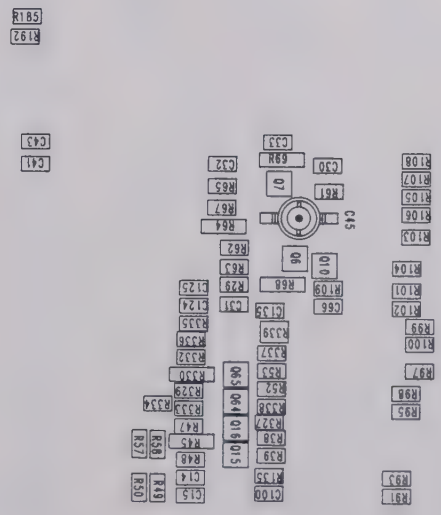
Main Board Schematic (19911-D-SD, sheet 3 of 3) 81

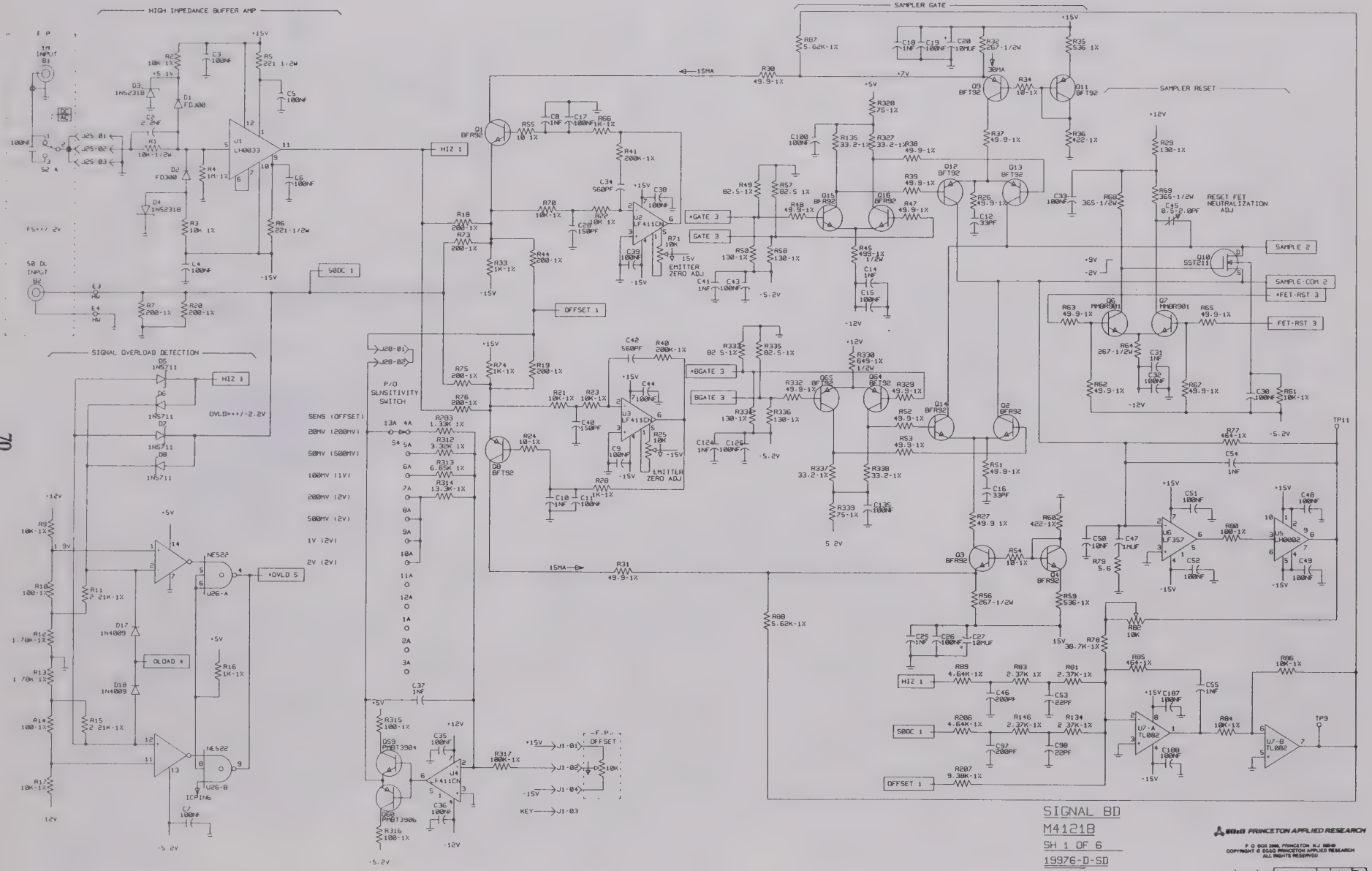
Display Board Layout Diagram 82

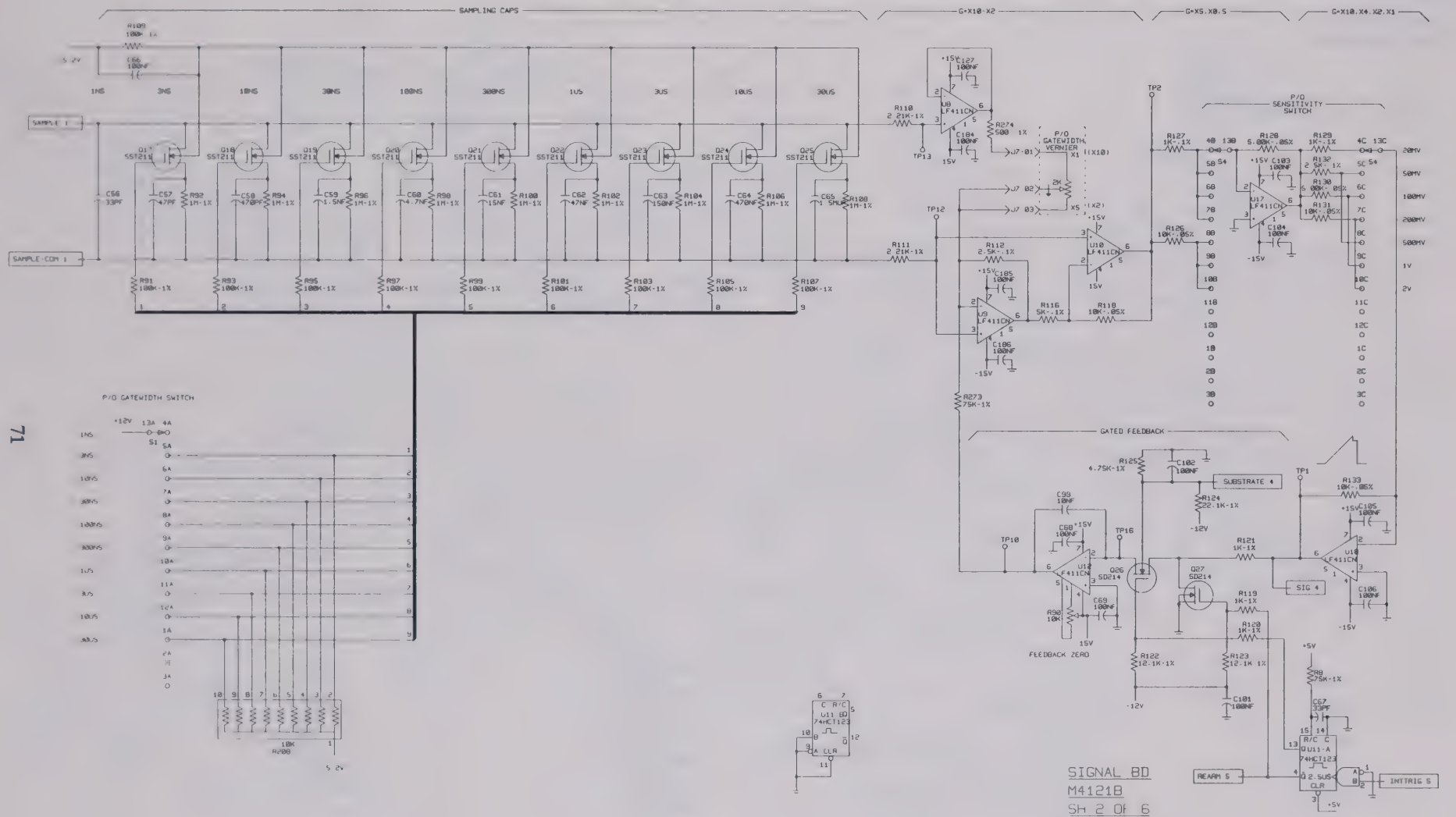
Display Board Schematic (19906-C-SD) 83


O' RING 3 PLACES
#2504-0088-0
TH1
TH2
ASSEMBLTY TOP SIDE

ASSEMBLY BACK SIDE

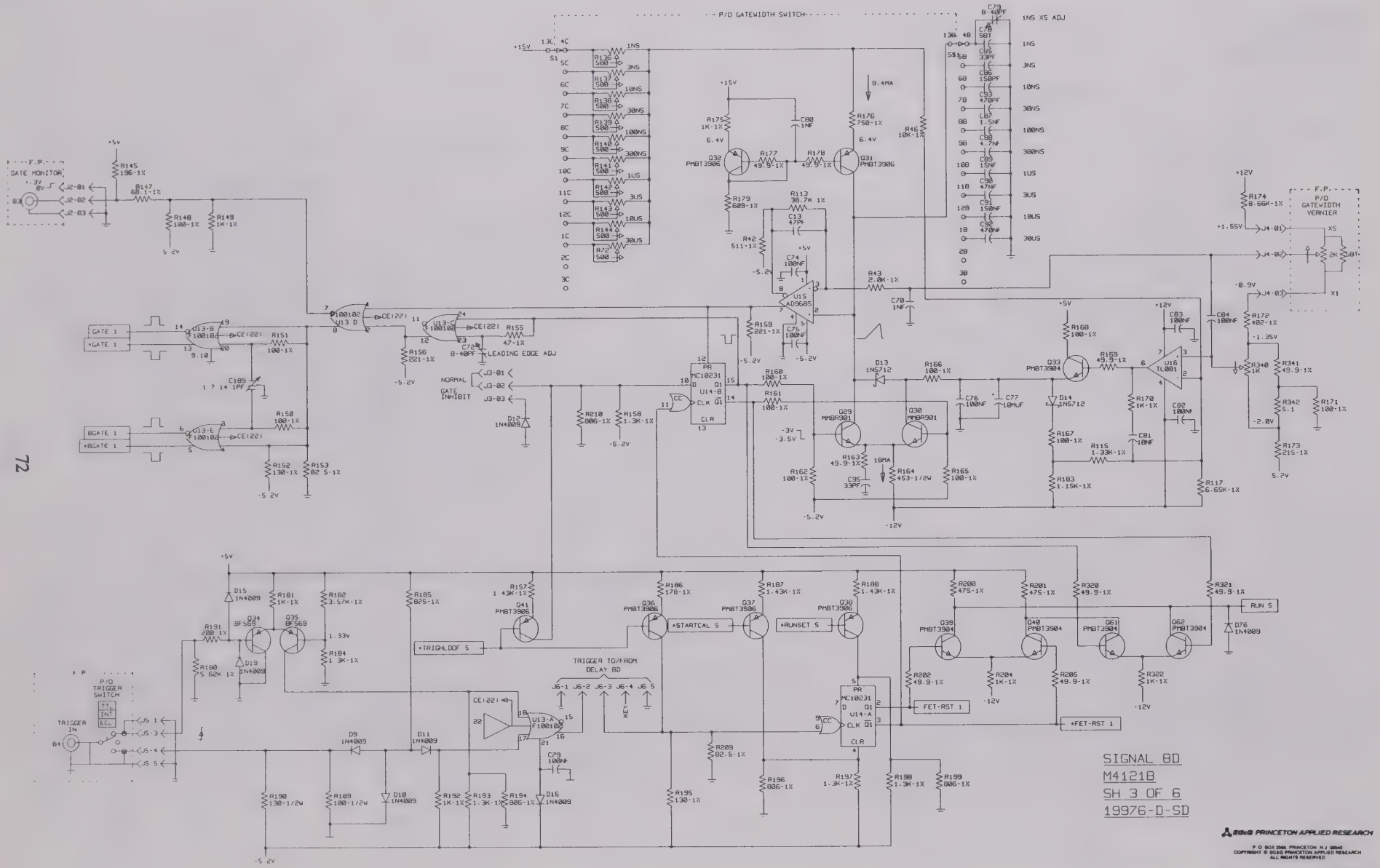


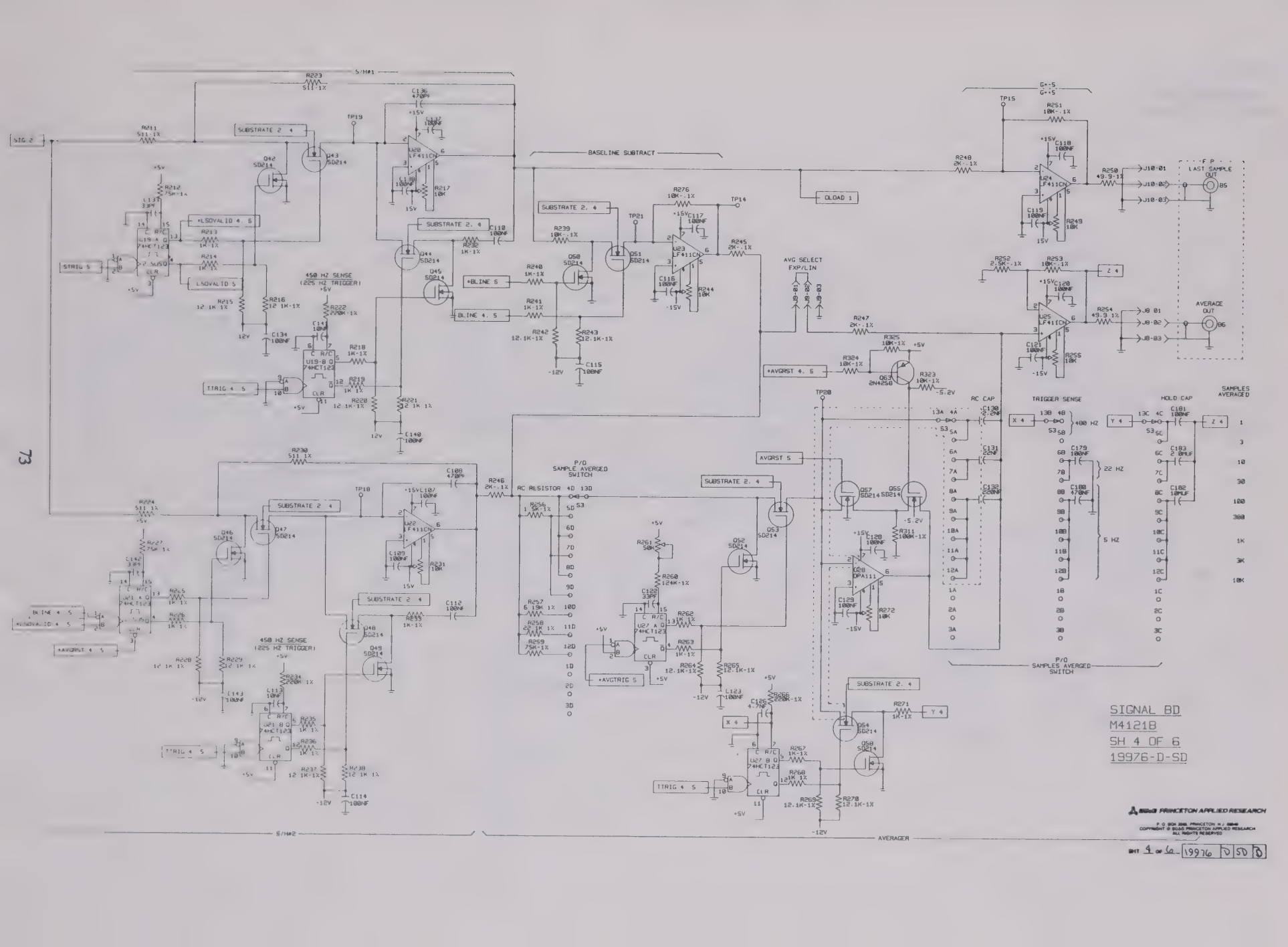

HIGH IMPEDANCE BUFFER AMP
SAMPLER GATE
SAMPLER RESET
SIGNAL OVERLOAD DETECTION
RESET FET NEUTRALIZATION ADJ
EMITTER ZERO ADJ
P/O SENSITIVITY SWITCH
SENS (OFFSET)
20MV (200MV)
50MV (500MV)
100MV (1V)
200MV (2V)
500MV (2V)
1V (2V)
2V (2V)
SAMPLE 2
SAMPLE-COM 2
*FET-RST 3
FET-RST 3
*GATE 3
GATE 3
*BGATE 3
BGATE 3
HI Z 1
OFFSET 1
50DC 1
*OVLD S
OLOAD 1
KEY
SIGNAL BD
M4121B
SH 1 OF 6
19976-D-SD
PRINCETON APPLIED RESEARCH
BMT. 1 OF 6 19976 D 50B

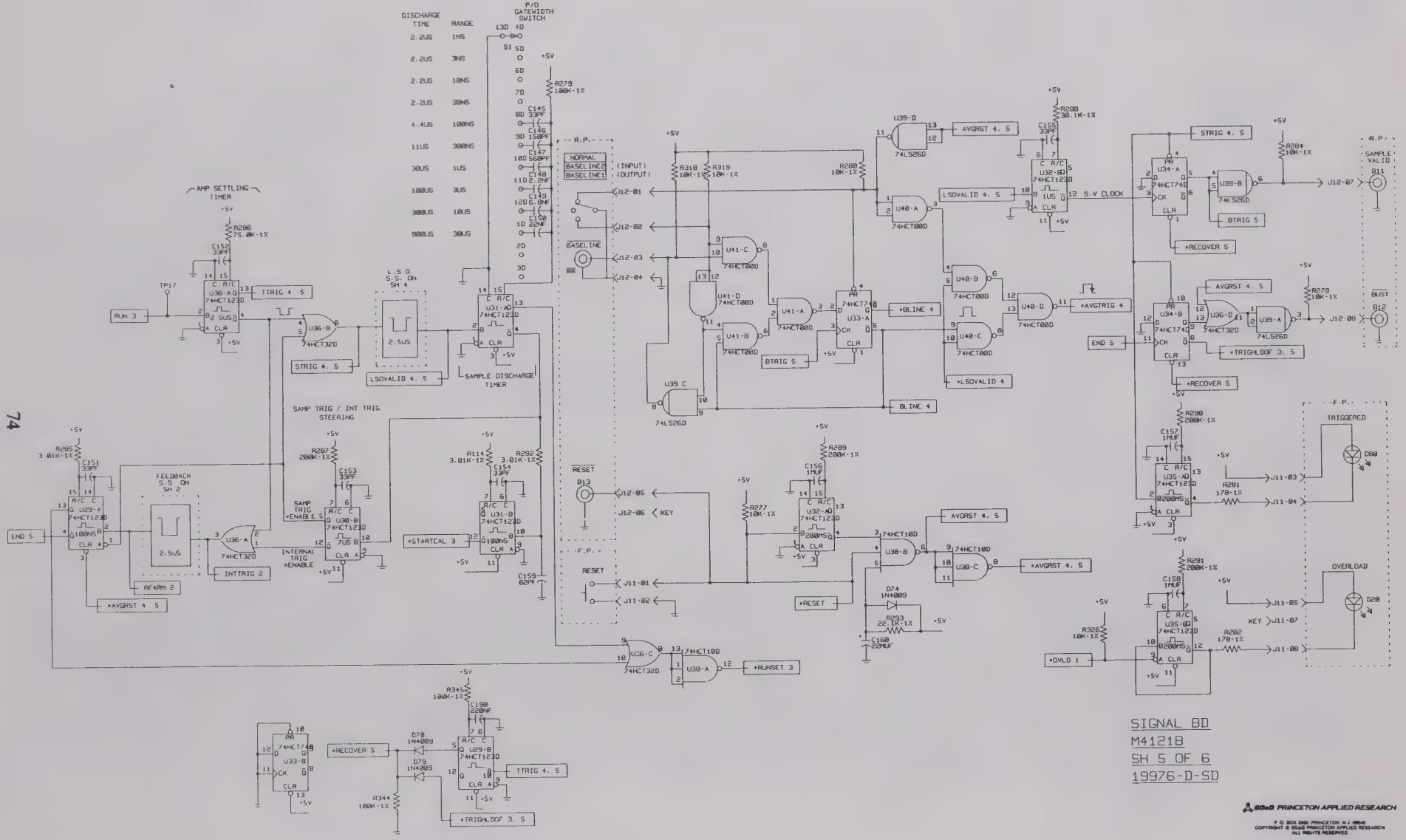


PRINCETON APPLIED RESEARCH
COPYRIGHT © PRINCETON APPLIED RESEARCH
ALL RIGHTS RESERVED

SHT. 2 OF 6 | 19976 | D | SD | B


P/O GATEWIDTH SWITCH
GATE MONITOR
F.P.
P/O GATEWIDTH VERNIER
NORMAL
GATE INHIBIT
LEADING EDGE ADJ
TRIGGER TO/FROM DELAY BD
P/O TRIGGER SWITCH
TRIGGER IN
*TRIG+LDOF S
*STARTCAL S
*RUNSET S
RUN S
*FET-RST 1
FET-RST 1
SIGNAL BD
M4121B
SH 3 OF 6
19976-D-SD
PRINCETON APPLIED RESEARCH
COPYRIGHT © EG&G PRINCETON APPLIED RESEARCH
ALL RIGHTS RESERVED
SHT. 3 OF 6 .9976 D SD B

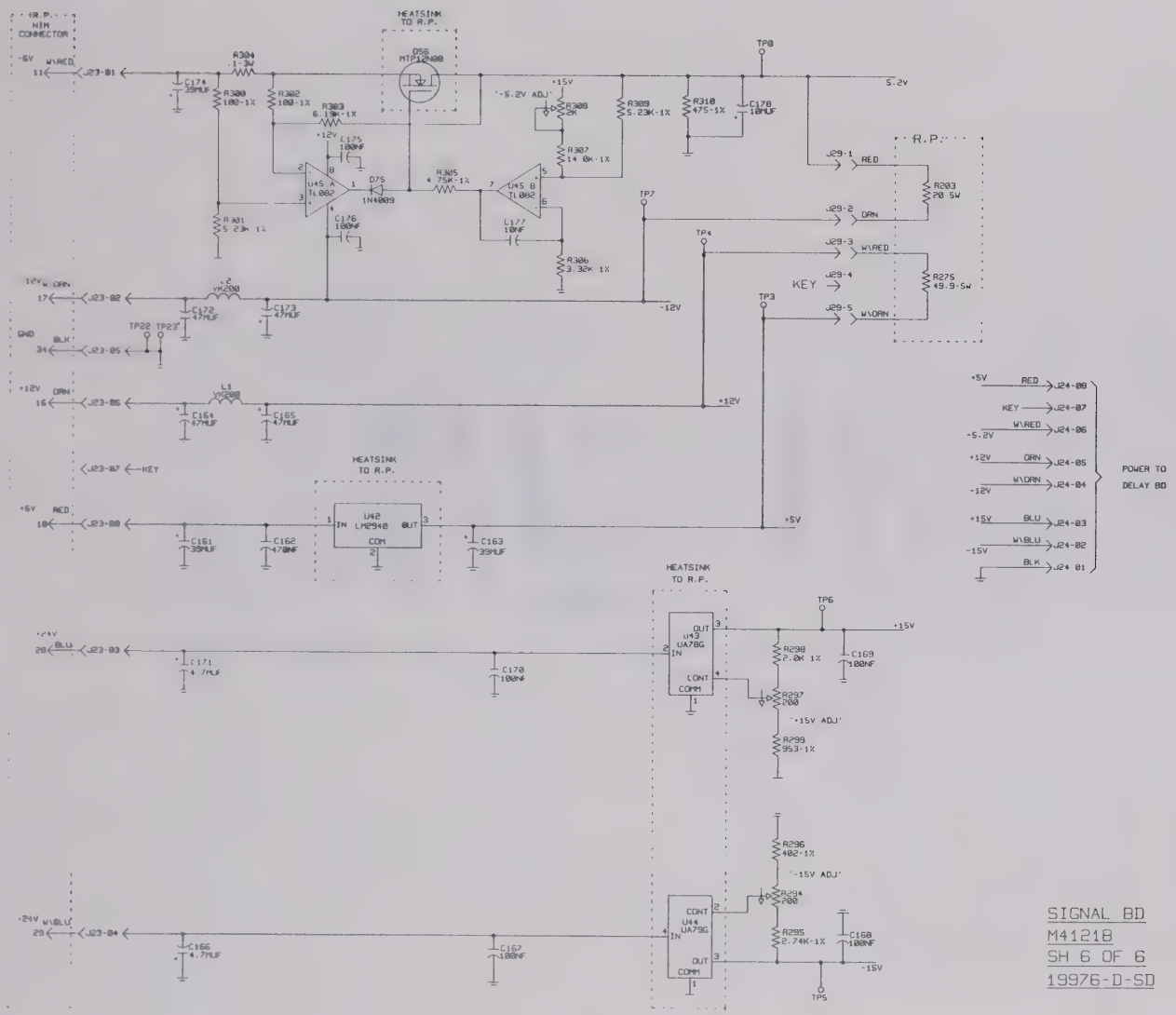

SIGNAL BD
M4121B
SH 4 OF 6
19976-D-SD
PRINCETON APPLIED RESEARCH
BASELINE SUBTRACT
AVERAGER
S/H#1
S/H#2
SUBSTRATE 2. 4
SIG 2
STRIG 5
TTRIG 4. 5
*LSOVALID 4. 5
LSOVALID 5
*BLINE 5
BLINE 4. 5
*AVGRST 4. 5
AVGRST 5
*AVGTRIG 5
450 HZ SENSE
(225 HZ TRIGGER)
P/O
SAMPLE AVERGED
SWITCH
AC RESISTOR
OLOAD 1
AVG SELECT
FXP/LIN
LAST SAMPLE
OUT
AVERAGE
OUT
RC CAP
TRIGGER SENSE
HOLD CAP
SAMPLES
AVERAGED
400 HZ
22 HZ
5 HZ
X 4
Y 4
Z 4
G=5
G=5



SIGNAL BD
M4121B
SH 5 OF 6
19976-D-SD

PRINCETON APPLIED RESEARCH
P.O. BOX 2565 PRINCETON N.J. 08540
COPYRIGHT © 1980 PRINCETON APPLIED RESEARCH
ALL RIGHTS RESERVED

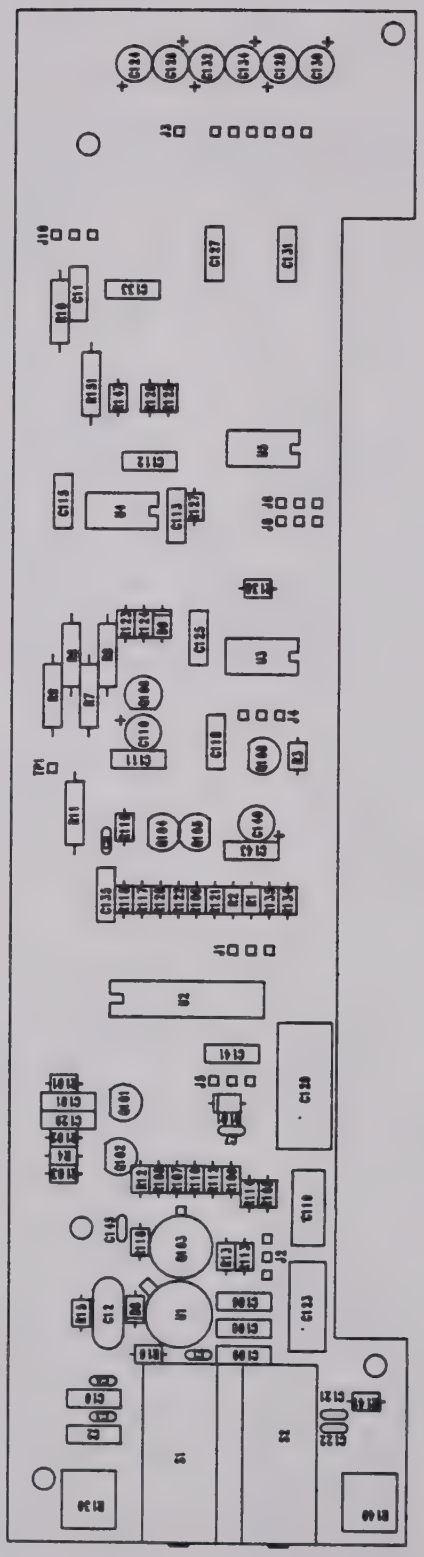
SHT. 5 OF 6 | 19976 | D | SD | B

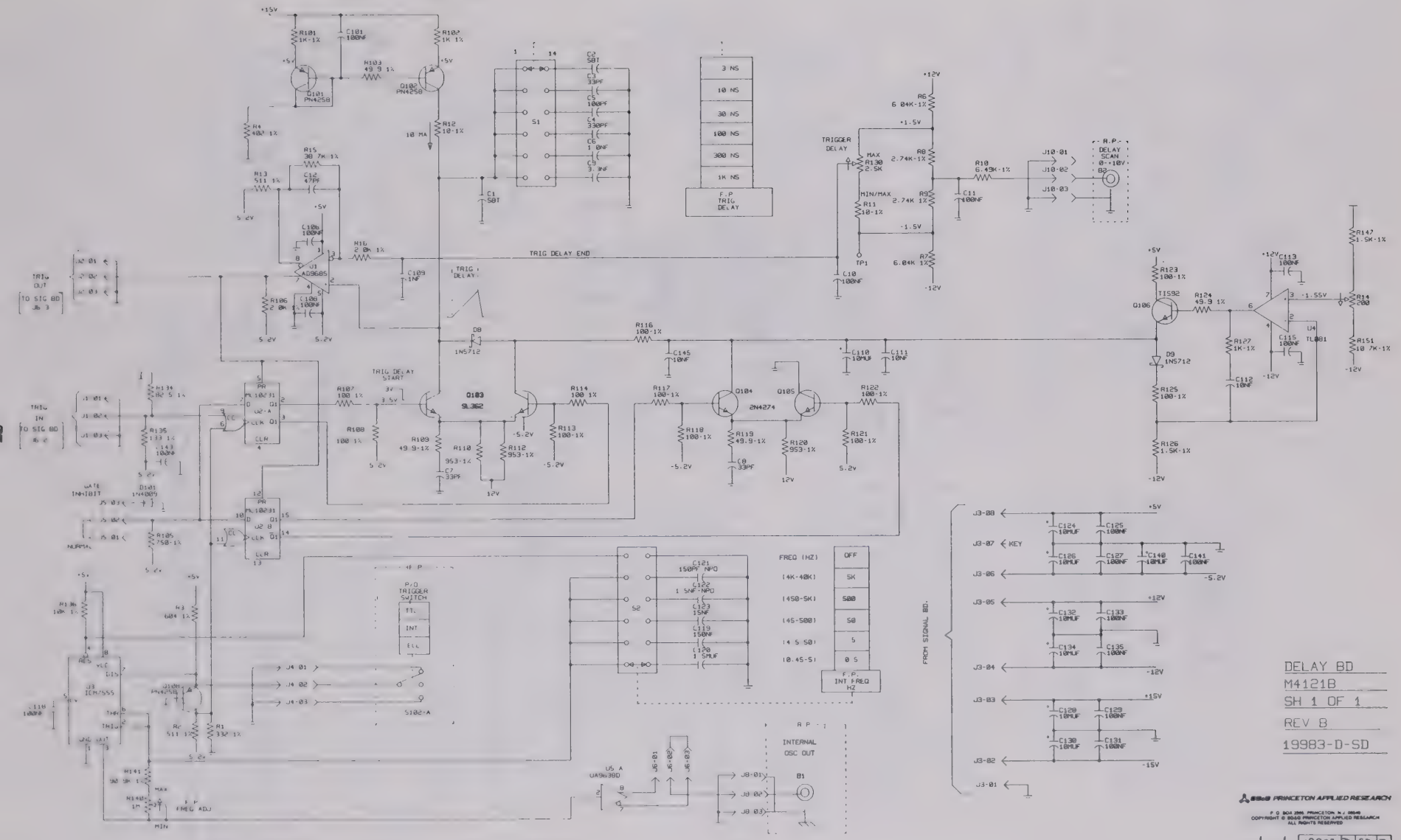


SIGNAL BD
M41218
SH 6 OF 6
19976-D-SD

PRINCETON APPLIED RESEARCH
P O BOX 2565, PRINCETON, N J 08540
COPYRIGHT © 1980 PRINCETON APPLIED RESEARCH
ALL RIGHTS RESERVED

SHT. 6 OF 6 | 19976 | D | SD | B

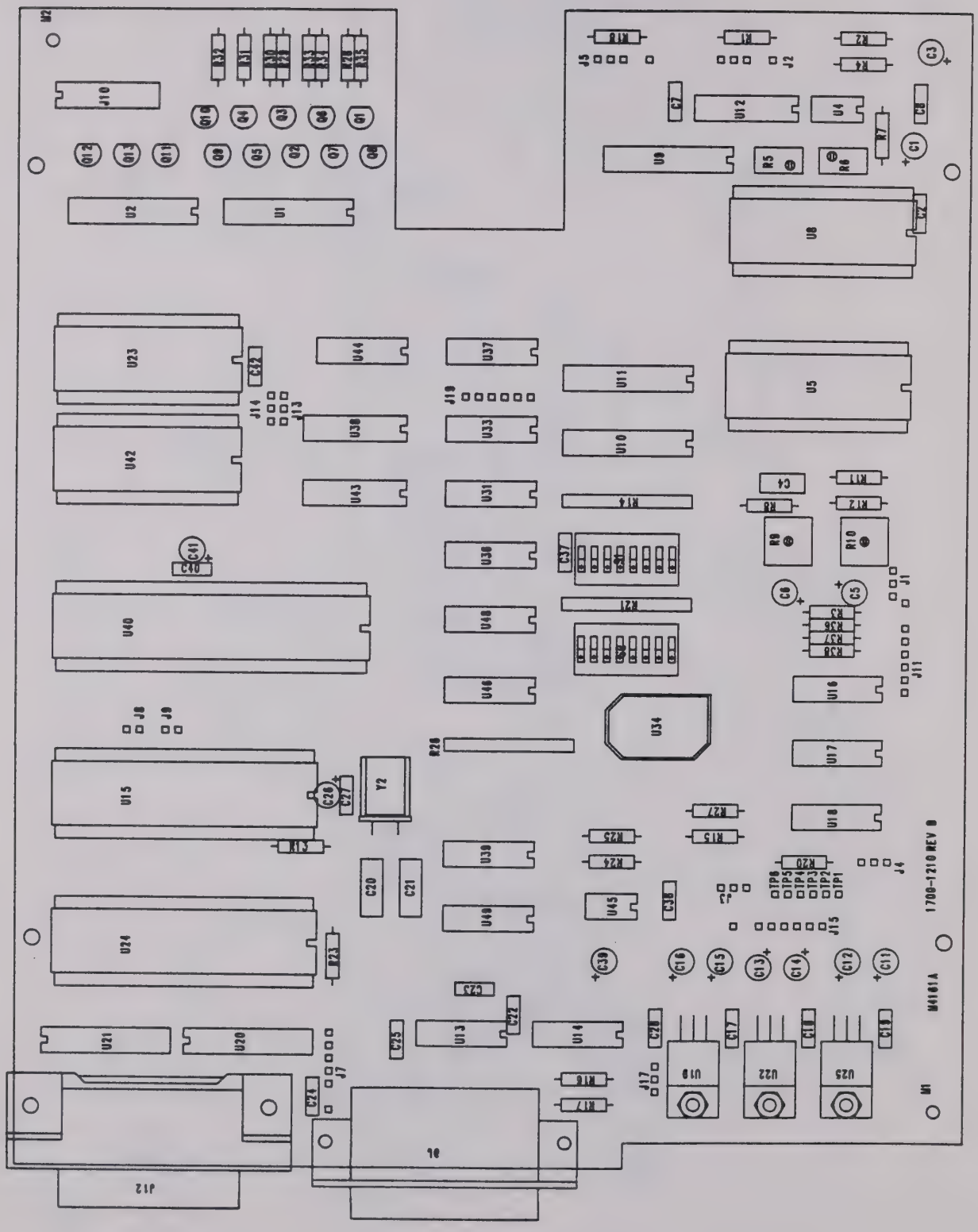


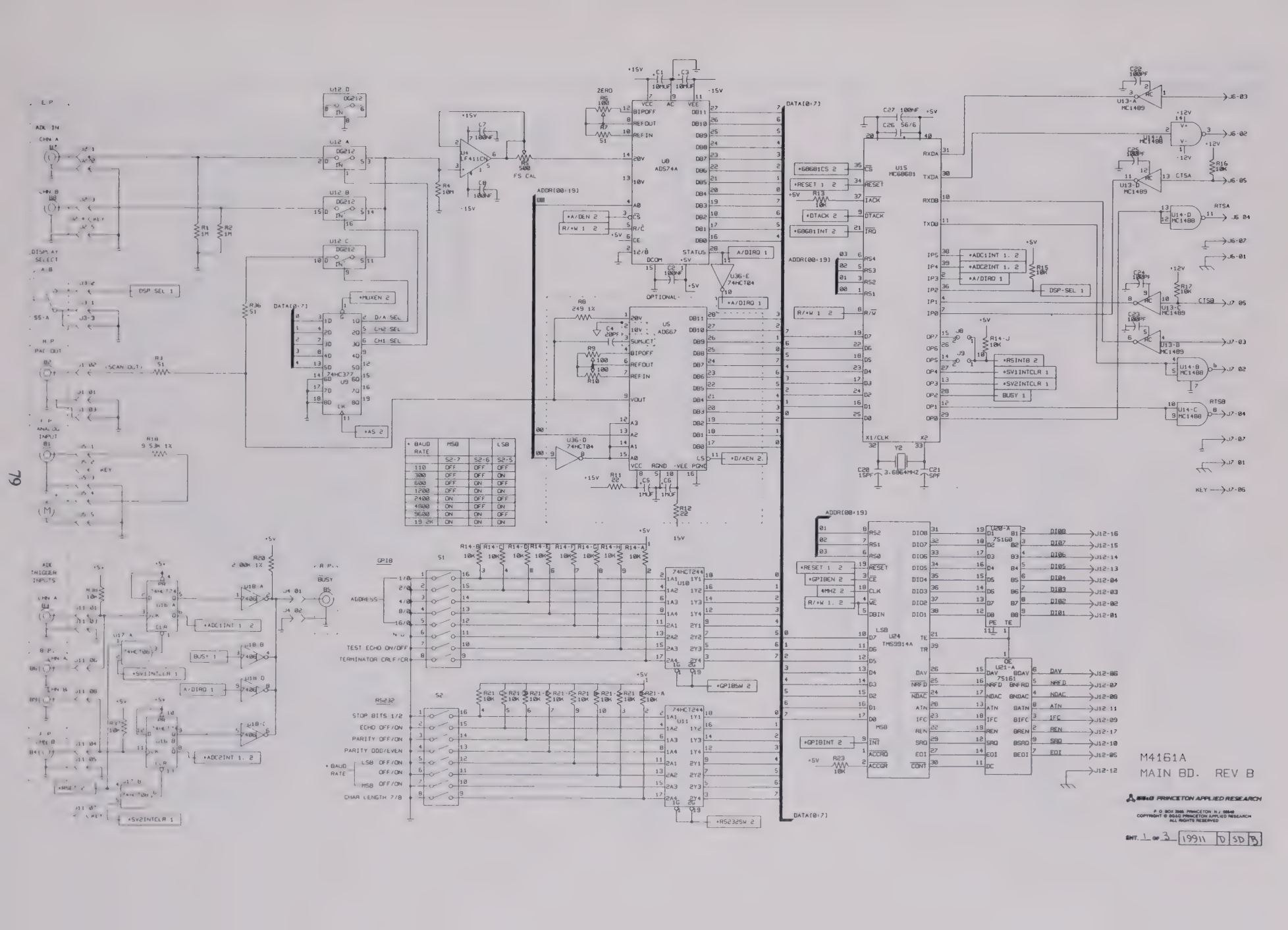


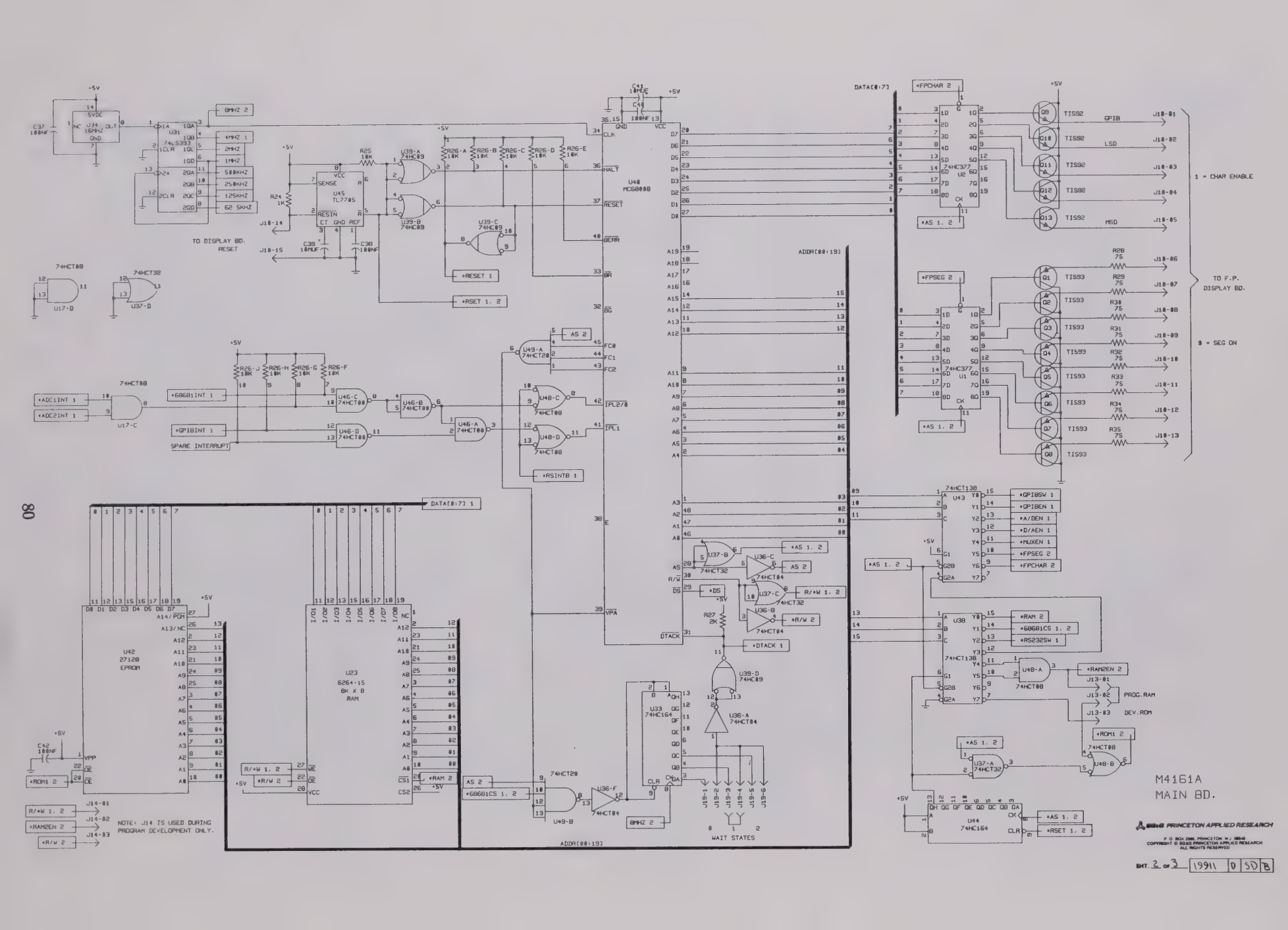
DELAY BD
M4121B
SH 1 OF 1
REV B
19983-D-SD

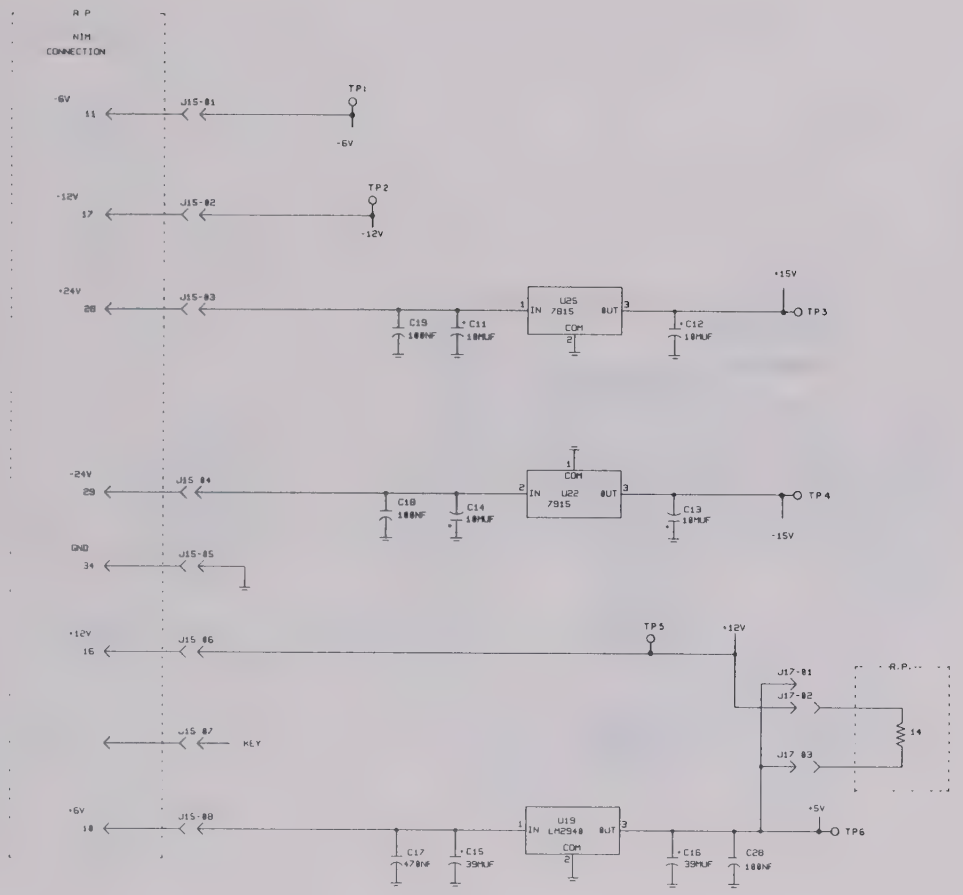
PRINCETON APPLIED RESEARCH
P O BOX 2565 PRINCETON N J 08540
COPYRIGHT © 1980 PRINCETON APPLIED RESEARCH
ALL RIGHTS RESERVED

SHT. 1 OF 1 | 19983 D SD B


1700-1210 REV B
M1
M2


M4161A
MAIN BD. REV B
PRINCETON APPLIED RESEARCH
P O BOX 2565 PRINCETON N.J. 08540
COPYRIGHT © 1985 PRINCETON APPLIED RESEARCH
ALL RIGHTS RESERVED
SHT. 1 OF 3 19911 D SD B
DSP SEL 1
*MUXEN 2
D/A SEL
CH2 SEL
CH1 SEL
*AS 2
*SCAN OUT
*D/AEN 2
A/DIRQ 1
*A/DEN 2
R/*W 1 2
*68681CS 2
*RESET 1 2
*DTACK 2
*68681INT 2
*ADC1INT 1. 2
*ADC2INT 1. 2
*RSINTB 2
*SV1INTCLR 1
*SV2INTCLR 1
BUSY 1
*GPIBEN 2
4MHZ 2
*GPIBINT 2
*GP10SW 2
*RS232SW 2
ADDR(00:19)
DATA(0:7)
BAUD RATE
ADDRESS
TEST ECHO ON/OFF
TERMINATOR CRLF/CR
STOP BITS 1/2
ECHO OFF/ON
PARITY OFF/ON
PARITY ODD/EVEN
CHAR LENGTH 7/8
ADC INPUTS
DISPLAY SELECT
ANA. OUT
ZERO
FS CAL
OPTIONAL
J6-03
J6-02
J6-05
J6-04
J6-07
J6-01
J7-05
J7-03
J7-02
J7-04
J7-07
J7 01
KEY
J7-06
J12-16
J12-15
J12-14
J12-13
J12-04
J12-03
J12-02
J12-01
J12-06
J12-07
J12-08
J12 11
J12-09
J12-17
J12-10
J12-05
J12-12

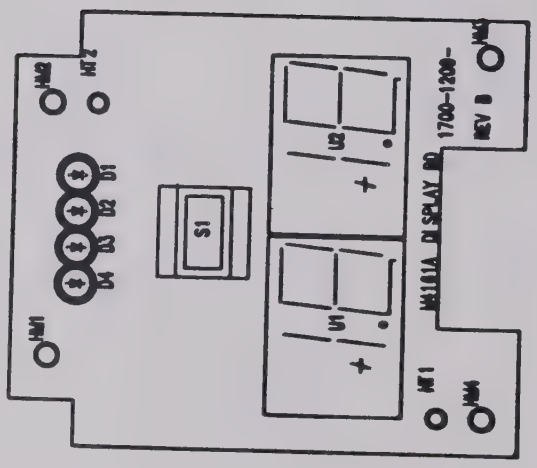

M4161A
MAIN BD.
PRINCETON APPLIED RESEARCH
P O BOX 2565, PRINCETON, N J 08540
COPYRIGHT © 1980 PRINCETON APPLIED RESEARCH
ALL RIGHTS RESERVED
SHT. 2 OF 3
19911 D SD B
TO DISPLAY BD. RESET
SPARE INTERRUPT
TO F.P. DISPLAY BD.
1 = CHAR ENABLE
0 = SEG ON
NOTE: J14 IS USED DURING PROGRAM DEVELOPMENT ONLY.
WAIT STATES
PROG. RAM
DEV. ROM
U42 27128 EPROM
U23 6264-15 8K X 8 RAM
U48 MC68008
U45 TL7705
U42
U23
U48
U45

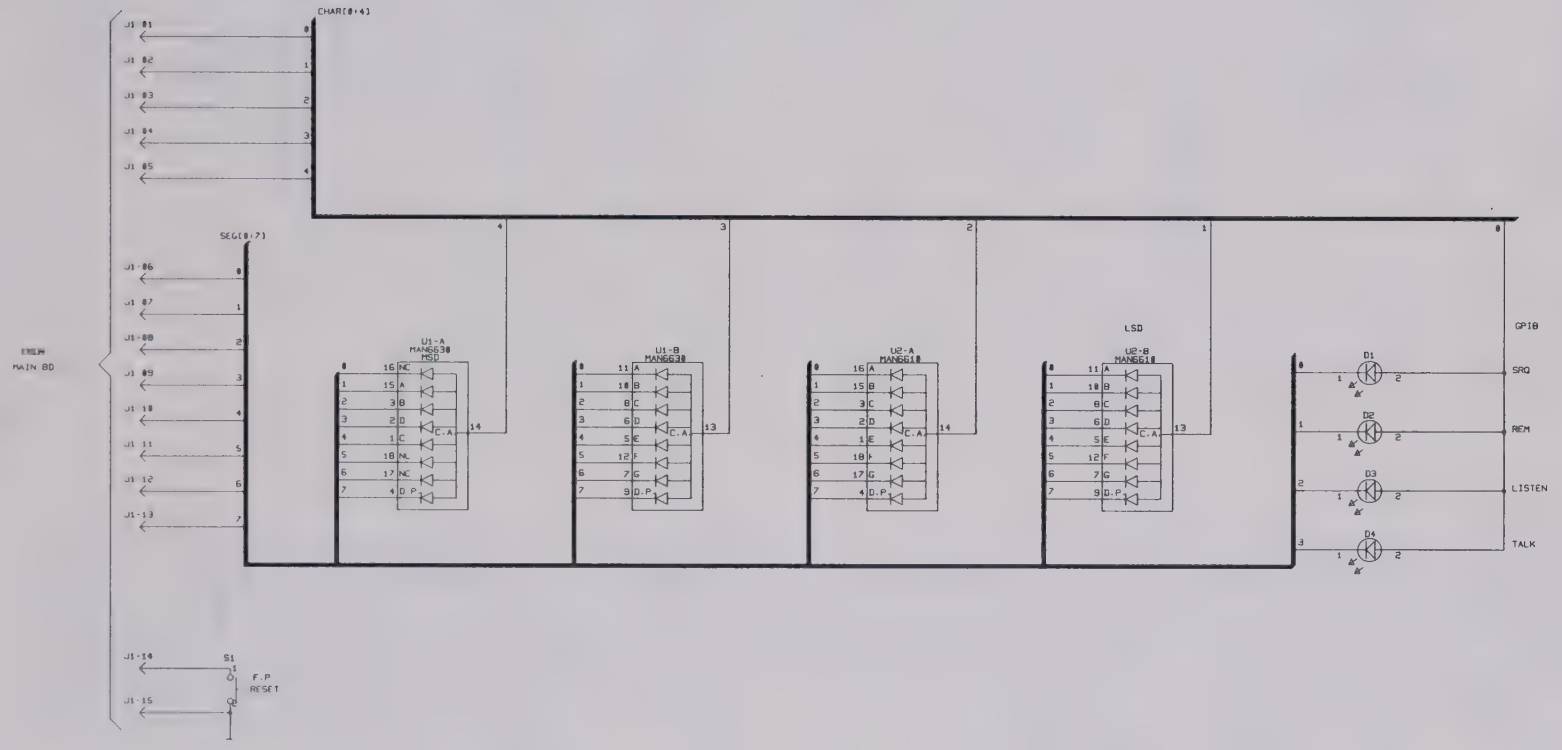


M4161A
MAIN BD.

PRINCETON APPLIED RESEARCH
P O BOX 2565 PRINCETON N J 08540
COPYRIGHT © 1986 PRINCETON APPLIED RESEARCH
ALL RIGHTS RESERVED

SHT 3 OF 3 | 19911 | D | SD | A


MH2
MH4
MH1
MH3
D4
D3
D2
D1
S1
U2
U1
1700-1200-
REV B
MK101A_DI_RELAY_BD



M4161A
DISPLAY BD.
19906-C-SD
SH. 1/1
REV B

Index

4100
- Description 1
- Features of 1
- System configurations 2
- System Connections 2

4100-1 2
4100-2 2
4100-3 2
4121B
- Averaging Techniques 7
- AVG connector 8
- Baseline I/O connector 11
- Bin Requirements 7
- BUSY OUT 10
- Circuit Descriptions 15
- Connections 11
- Coupling 9
- Delay Scan In connector 10, 11
- Delay Vernier & Switch 10
- Description of 5
- Front Panel 8
- Gate Mon 8
- Gate Monitor connector 9
- Gate Width 7, 12
- Gate Width Multiplier 8
- Gate Width Range 8
- Grounding 8
- Input Coupling 9
- Inputs 9
- Installation 7
- INT OSC OUT connector 10
- LSO connector 8
- Offset dial 8
- Output 12
- OVLD Indicator 8
- Rear Panel 10
- Reset Avg I/O connector 10
- Reset pushbutton 8
- SAMPLE VALID OUT 10
- Samples Averaged switch 8
- Sampling 7
- Sampling Mode switch 11
- Schematics, table of 67
- Sensitivity Selector 8
- Signal Channel 8
- Signal Sensitivity 12
- Specifications 5
- System Connections 11
- Timing and Control 6
- Trigger Indicator 9
- Trigger Input connector 9
- Trigger Mode Selector 9
- Trigger Source 6
- Triggering, External 6
- Triggering, Internal 6

4144
- * key 21
- A and B Outputs 20
- A B AB 23
- A Connector 22
- A.B connector 22
- Applications 19
- B Connector 22
- Burst Mode 20
- CH A DELAY 22
- CH B DELAY 23
- Code Sequences 25
- Command Syntax 25
- Commands 26
- Cursor 21
- Delay Accuracy 20
- Delay Range 20
- Dimensions 21
- Display 21
- Display Functions 22
- Display Messages 22
- Ext/Stop/Int switch 22
- External Triggering 23
- Features 19
- Features of 1
- Fixed Delay Operation 24
- Function keys 21
- GPIB ADDRESS 23
- GPIB Functions Supported 21
- GPIB Interface 25
- GPIB Serial Poll 26
- Introduction to 19
- Keypad 21
- LINE Trigger mode 22
- Numeric keys 21
- Operation 21
- Output Pulse Levels 20
- Outputs 20
- Pen Lift 22
- PEN LIFT DELAY IN/OUT 23
- Power Requirements 21
- Ramp Out 20
- Ramp Output 22
- RATE 23
- Rear Panel 22
- RUN SCAN 23
- SAVE/RECALL 23
- Save/Recall Parameters 21
- SCAN INCREMENT 23
- Scan Inhibit 20
- Scan Inhibit connector 22
- Scan Mode 20
- Scanned Operation 24
- Single Shot Triggering 24
- Specifications 20
- Static Interface Routine 43
- STEPS/SCAN 23
- Syntax 25
- T display message 23
- Trigger Characteristics 20
- Trigger Impedance Select switch 22
- Trigger Inhibit 22
- Trigger Input connector 22
- TRIGGER LEVEL 23

Trigger Mode 22
Trigger Modes 20
Triggered Indicator 22
Triggering 23
Triggering When Scanning 24
TRIGGERS/STEP 23
Typical Operation 25
Width Adjustments 22
4152 2, 7, 8, 28
4152/99 2, 7, 28
4161A
A/D Converter 36
ADC Inputs 28
ADC Signal Inputs 27
ADC Trigger Inputs 27, 28
Analog Meter 28
Baud Rate Selection 33
Bin Requirements 28
BSYON 30
BUSY OUT 29
Circuit Descriptions 35
Command Overruns & Serial Polling 31
Commands 30
Computer Control 30
Data Format 28
Data Rate 28
Display 28
Display Select switch 28
Front Panel 28
GPIB Address switch 30
GPIB connector 29
GPIB Indicators 28
Grounding 28
Installation 28
Introduction to 27
Logic Levels 34
NIM connector 29
Operation 29
Rear Panel 29
Reset pushbutton 29
RS232C Command Overruns 34
RS232C Connector 29
RS232C Echo 33
RS232C Handshake 33
RS232C Interface 33
RS232C Logic Levels 34
RS232C Parameter Selection 32
RS232C Pinout 60
RS232C Transmit/Receive Pin Assignments 33
RS232C Troubleshooting 34
Schematics, table of 67
Serial Poll Status Byte 32
Serial Polling 31
Specifications 27
Static Interface Routine 55
Synchronized Operation 30
Terminator requirements 30
Test Echo 30
Ventilation 28
7018 2, 7, 28
A.B 20, 22, 24
ADC 3, 11, 12, 27-29
ASCII Character Codes 63
ATN 37
Averager Output 7
Averaging Techniques 7
AVG 7-12, 17
AVG connector 7, 9, 12
Avg. Droop Rate 7
Baseline 1 7, 11, 18
Baseline 2 7, 11, 18
BASELINE I/O 7, 11, 18
Baud Rate 33-36, 39, 40
Bin 1, 2, 7, 8, 27, 28, 44
BIN 1 Command 31
BIN0 Command 32
BSYOFF 31
Burst 19-21, 24, 25
Burst Mode 19, 20, 25
BUSY 3, 9-12, 18, 28-32, 34, 36
BUSY OUT 12, 29
CLEAR-TO-SEND 33-35, 40
Command Done 32
CSF 31
DCE 33, 34, 39, 40
Delay
Trigger to Gate 6
DELAY SCAN IN (0 - +10 V) 11
DEVICE CLEAR 37
DSPOFF Command 32
DSPON Command 32
DTE 33, 34, 39, 40, 60
Echo 30, 33, 35, 36
ECL 6, 7, 9-12, 16-18, 20, 22, 23
EOI 37
Exponential Averaging 1, 7, 8, 12, 17
FCC Notice 61
Front Panel
Power On/Off switch 21
GATE MON 7-9, 11, 12
Gate Monitor 8, 9
Gate Monitor Output 7
Gate Width 7
Gate Width Multiplier 8
Gate Width Range 7, 8
Gate Width Vernier 7
GPIB 1, 3, 19, 21-37, 43-48, 51-53, 56-58-, 59
ATN 37
EOI 37
GENERAL INTERFACE MANAGEMENT LINES 37
IFC 37
Pinouts 59
REMOTE MESSAGES 37
REN 37
SRQ 37
GPIB Background 37
GPIB REMOTE MESSAGES
DCL 37

LISTEN 37
SPD 37
SPE 37
TALK 37
UNLISTEN 37
UNTALK 37
Grounding 8, 28
IEEE-488 1, 3, 21, 22, 25, 27-29, 56, 59
IFC 37
intrinsic delay 6, 8, 10, 11
Jumper 6-8, 11, 12, 15-18, 35, 56
JUMPERS 12
Averaging Mode 12
Delay Bypass 6, 15
Internal Osc. Polarity 15
Last Sample 1, 3, 7
Linear Averaging 5, 7, 8, 12, 17
LISTEN 27, 28, 37, 45, 46
LSO 3, 7-9, 11, 12, 16, 18, 28
LSO connector 9, 12, 28
NIM 1, 5, 7, 10, 11, 19, 20, 22-24, 27-29-, 60
NIM connector pinout 60
Output Ready 32, 58
OVLD 8, 16
PC2 3
PC2A 3, 25, 56
PEN LIFT 20
Pinouts 59
Power Requirements
4121B 2
4144 21
4161A 2
RAMP OUT 20
RDA 30
RDAB 30
RDB 30
REM 28, 37, 44
REN 37
REQUEST-TO-SEND 33-35, 40
RESET 7, 8, 10, 12, 16-18, 23, 27, 29, 35, 58
RS232 36, 60
RS232C 1, 27-30, 32-35, 37, 39-41, 60
History & Problems of 39
Introduction to 39
Pinout 60
Word Construction 39
RS232C Background 39
SA, SB, & SAB 31
Sample Valid 10-12, 18
SAMPLE VALID OUT 11, 12
Sampling 7
Baseline 1 7
Baseline 2 7
Normal 7
Scan Inhibit 20, 22
Scan Mode 19-21, 25
Schematics, table of 67
SERIAL POLL ENABLE and SERIAL POLL DISABLE 37
Serial Polling 32, 37
SNIR 5, 8, 12
Specifications
4121B 5
SRQ 27, 28, 32, 37, 44, 59
SSF 31
SV 3, 12
TALK 27, 28, 37, 45-47
TALK, UNTALK, LISTEN, and UNLISTEN 37
TID-20893 7, 28
To 20
Trigger Active Indicator 6
Trigger Inhibit 3, 18, 20, 22, 26, 28, 29
TRIGGERING 3, 5-10, 12, 15, 17, 22-24-, 25, 28
TTL 6-12, 17, 18, 20, 22, 23, 27-29, 35, 36, 40
X Command 31
XON/XOFF 33
Y Command 31

WARRANTY

EG&G Instruments Corporation warrants each instrument of its own manufacture to be free of defects in material and workmanship. Obligations under this Warranty shall be limited to replacing, repairing or giving credit for the purchase price, at our option, of any instrument returned, shipment prepaid, to our Service Department for that purpose within **TWO** years of delivery to the original purchaser, provided prior authorization for such return has been given by an authorized representative of EG&G Instruments Corporation.

This Warranty shall not apply to any instrument, which our inspection shall disclose to our satisfaction, to have become defective or unworkable due to abuse, mishandling, misuse, accident, alteration, negligence, improper installation, or other causes beyond our control. This Warranty shall not apply to any instrument or component not manufactured by EG&G Instruments Corporation. When products manufactured by others are included in EG&G Instruments Corporation equipment, the original manufacturer's warranty is extended to EG&G Instruments Corporation's customers.

EG&G Instruments Corporation reserves the right to make changes in design at any time without incurring any obligation to install same on units previously purchased.

THERE ARE NO WARRANTIES WHICH EXTEND BEYOND THE DESCRIPTION ON THE FACE HEREOF. THIS WARRANTY IS IN LIEU OF, AND EXCLUDES ANY AND ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESSED, IMPLIED OR STATUTORY, INCLUDING MERCHANTABILITY AND FITNESS, AS WELL AS ANY AND ALL OTHER OBLIGATIONS OR LIABILITIES OF EG&G INSTRUMENTS CORPORATION, INCLUDING, BUT NOT LIMITED TO, SPECIAL OR CONSEQUENTIAL DAMAGES. NO PERSON, FIRM OR CORPORATION IS AUTHORIZED TO ASSUME FOR EG&G INSTRUMENTS CORPORATION ANY ADDITIONAL OBLIGATION OR LIABILITY NOT EXPRESSLY PROVIDED FOR HEREIN EXCEPT IN WRITING DULY EXECUTED BY AN OFFICER OF EG&G INSTRUMENTS CORPORATION.

SHOULD YOUR EQUIPMENT REQUIRE SERVICE

A. Contact the Service Department (609-530-1000) or your local representative to discuss the problem. In many cases it will be possible to expedite servicing by localizing the problem to a particular plug-in circuit board.

B. If it is necessary to send any equipment back for service, we need the following information.

1. Model number and serial number.
2. Your name (instrument user).
3. Your address.
4. Address to which the instrument should be returned.
5. Your telephone number and extension.
6. Symptoms (in detail, including control settings).
7. Your purchase order number for repair charges (does not apply to repairs in warranty).
8. Shipping instructions (if you wish to authorize shipment by any method other than normal surface transportation.)

C. **U.S. CUSTOMERS** - Ship the equipment being returned to:

EG&G PARC
375 Phillips Blvd.
Trenton, NJ 08618

D. **CUSTOMERS OUTSIDE OF U.S.A.** - To avoid delay in customs clearance of equipment being returned, please contact the factory or the nearest factory distributor for complete shipping information.

E. Address correspondence to:

EG&G PARC
P.O. Box 2565
Princeton, New Jersey 08543-2565

Phone: (609) 530-1000
TELEX: 84-3409
FAX: 883-7259

www.manualsplus.com

1024888

EG&G
PRINCETON APPLIED RESEARCH
P.O. BOX 2565 • PRINCETON, NJ 08543-2565, USA • 609/452-2111 • TELEX: 843409